Chapter 16
Security, Privacy, and Usability Challenges in Selfie Biometrics

Mikhail Gofman, Sinjini Mitra, Yu Bai and Yoonsuk Choi

Abstract From biometric image acquisition to matching to decision making, designing a selfie biometric system is riddled with security, privacy, and usability challenges. In this chapter, we provide a discussion of some of these challenges, examine some real-world examples, and discuss both existing solutions and potential new solutions. The majority of these issues will be discussed in the context of mobile devices, as they comprise a major platform for selfie biometrics; face, voice, and fingerprint biometric modalities are the most popular modalities used with mobile devices.

16.1 Introduction

Modern mobile devices support face, voice, fingerprint, and iris recognition. These biometric systems operate under uncontrolled conditions; they must contend with security threats of fake biometrics; they must protect against the divulgence of biometric templates if the device is lost or stolen; and they are constantly pressured to be user-friendly. In this chapter, we will provide an overview of security and usability challenges and solutions in mobile biometric systems. Special focus will be placed on issues of security attacks involving fake biometrics, template security, and the

M. Gofman (✉)
Department of Computer Science, California State University, Fullerton, 800 N State College Blvd, Fullerton, CA 92831, USA
e-mail: mgofman@fullerton.edu

S. Mitra
Department of Information Systems and Decision Sciences, California State University, Fullerton, 800 N State College Blvd, Fullerton, CA 92831, USA
e-mail: smitra@fullerton.edu

Y. Bai · Y. Choi
Department of Computer Engineering, 800 N State College Blvd, Fullerton, CA 92831, USA
e-mail: ybai@fullerton.edu

Y. Choi
e-mail: yochoi@fullerton.edu

A. Rattani et al. (eds.), *Selfie Biometrics*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-3-030-26972-2_16

making of mobile biometric systems user-friendly. The chapter concludes with the discussion of case studies concerning selfie biometric systems.

16.2 Security Issues Overview

The iPhone 5s was among the first commercially successful consumer mobile devices that supported fingerprint recognition [1]. Within a week of its release (2013), Chaos Computer Club (CCC), a German hacker group, had bypassed the fingerprint sensor "using easy everyday means." According to the CCC website:

> The fingerprint of the enrolled user is photographed with 2400 dpi resolution. The resulting image is then cleaned up, inverted and laser printed with 1200 dpi onto transparent sheet with a thick toner setting. Finally, pink latex milk or white wood glue is smeared into the pattern created by the toner onto the transparent sheet. After it cures, the thin latex sheet is lifted from the sheet, breathed on to make it a tiny bit moist and then placed onto the sensor to unlock the phone. [2]

Since then, methods were identified that could both bypass fingerprint recognition on later models of Apple iPhone [3]—and on Android-based devices [4]—and defeat face recognition systems on more modern devices, such as the iPhone X (released in 2017) [5, 6]. Attacks of this type are becoming progressively more sophisticated and effective as hackers continue to develop new methodologies. As an increasing number of users continue to ditch passwords and pin codes in favor of selfie biometric systems as their primary security gatekeepers, it is critical that these systems remain resilient to security attacks.

In addition to the security threats posed by fake biometric attacks, there exist concerns about the security and privacy of the data in the biometric templates. A template is a digital representation of the user's identifying features that are created from biometric samples initially supplied by the user when he/she sets up his/her device. The samples provided at the time of authentication are then matched against the stored template. If a template is divulged—say, if the device is lost, stolen, or hacked—hackers can use the template data to bypass biometric systems that use the same biometric modality.

The consequences of stolen biometric data are exacerbated by the fact that biometric modalities cannot be as easily changed as passwords can. Moreover, the stolen template data can be used for surveillance purposes in order to track users while they use the compromised biometric in different places and at different times.

These security concerns prompted mobile device manufacturers and security researchers to develop various software- and hardware-based defenses. To give the reader a better grasp of these security challenges and solutions, we begin with a generic threat model applicable to all biometric systems. We then focus on developing solutions to defend against trait-spoofing attacks and protect templates in mobile devices.

16.3 The Threat Model

The model depicted in Fig. 16.1 is adapted from Ratha et al. [7].

In this model, the *sensor* is used to acquire the raw biometric data. Next, the *feature extractor* module extracts the identifying information from the raw data. The features are then matched against the templates of enrolled users by the *matcher* module. Finally, the matcher outputs a "yes/no" decision as to whether the sample supplied during authentication matches the stored template.

The system in Fig. 16.1. The selfie biometric system threat model is susceptible to the following threats:

1. The attacker can place or present a *fake biometric* on the sensor or in front of the camera (e.g., a fake finger or a photograph of a face) in order to result in a false positive identification of an illegitimate subject. Multiple such attacks have proven to be successful against mobile devices [3–5].
2. Raw biometric data from the sensor can be *recorded* and *replayed* such that the attacker may gain access to the system (e.g., voice samples).
3. A *feature extractor may be replaced* by the attacker with another extractor that generates a predetermined set of features.
4. *Features extracted from biometric data can be replaced* with some other features chosen by the attacker.
5. A *trait-matching algorithm can be replaced* with the attacker's own matching algorithm.
6. *Biometric templates can be accessed by and tampered with* by the attacker. This includes insertion, deletion, modification, or theft of the templates.
7. *The retrieval of the template from the template database can be compromised*; for example, the attacker can replace the template retrieved from the requested user with his/her own template.

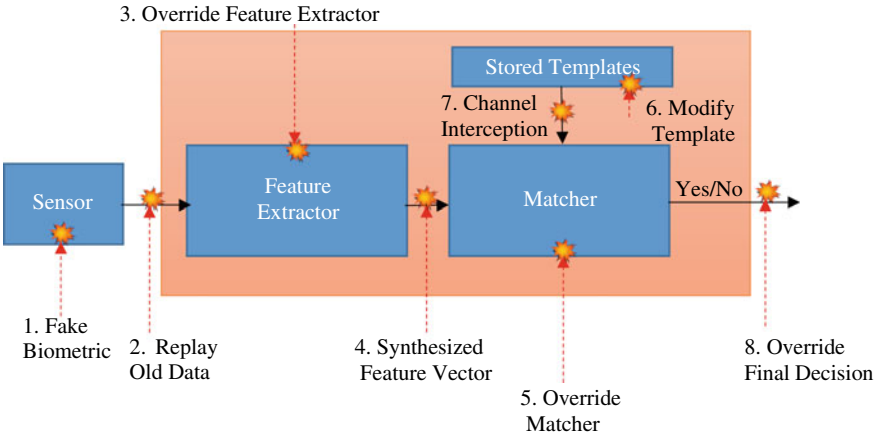

Fig. 16.1 The selfie biometric system threat model

8. *The decision of the identity verification system can be overridden* by the attacker such that he/she may gain unauthorized access to the system or deny access to a legitimate user.

Mobile device manufacturers have developed specialized computing hardware that helps mitigate attacks (2)–(8), such as the Apple Corporation's Secure Enclave processor [8] used with iPhones. This hardware is physically isolated from the main computing architecture of the device. Therefore, even if the applications, operating system, and the primary computing hardware are compromised, the function of the biometric system remains unaffected.

To further mitigate an attack (8), the templates store an output of a one-way function computed from the original biometric features. This output can still be used to match the features while rendering the derivation of the original identifying features computationally difficult or impossible. This is an important consideration in mobile devices because, even if the data is stored on a physically isolated, tamperproof hardware chip, attackers can disassemble a lost or stolen device in an attempt to bypass tamperproof hardware security mechanisms and thus retrieve the data.

Attack (1) remains an important concern. Some recent, noteworthy compromises of selfie biometric systems include:

- According to The Verge, "All it took was some dental mold to take a cast, some play-dough to fill it, and then a little trial and error to line up the play-dough on the fingerprint reader. We did it twice with the same print: once on an iPhone 6 and once on a Galaxy S6 Edge" [9].
- According to MyBroadband: "[When] our new gelatin [cast of person's fingerprints], was placed on the Nokia 5's sensor, the result was almost instant—the device was unlocked" [10].
- iPhone X's Face ID face recognition was bypassed by a Vietnamese company who created a 3-D mask of the individual's face that the device recognized as the face of the legitimate user [5].
- There were reports of children using their faces to unlock their parents' locked iPhone Xs using Face ID because children's faces may be sufficiently similar to their parents' faces [11].
- There were reports and demonstrations of iPhone X being unlocked by people who did not look alike [12].
- Banking mobile applications based on face recognition have been bypassed using a pre-recorded video of the user's face [13].

To help combat these and similar attacks, researchers and mobile device manufacturers have developed more robust sensors and additional hardware-based liveness testing techniques. These techniques help ensure the biometric reading from the sensor is indeed given by a living human being (e.g., checking the finger's pulse during fingerprint recognition and requiring eye blinking during face recognition).

A variety of software-based data processing techniques for detecting spoofed biometrics have also been proposed. Some have focused on frustrating attacks directed at specific modalities (e.g., face, voice, and fingerprint), while some proposed recognizing people based on multiple biometric modalities in order to challenge the attacker

to falsify more than one modality. Although these measures are useful, experience implies that all measures are likely to be eventually defeated by the attackers—the question is not "if," but rather, "when." Regardless, continuous innovation in technologies and methods for detecting fake biometrics is a practical necessity.

Next, we discuss attacks (1) and (7) along with their countermeasures. In selfie biometrics, these particular attacks have proven to be the most widely executed in practice, the most widely discussed in the literature, and the utmost focus of public concern.

16.3.1 Presentation Attacks

Throughout this section, we use the ISO/IEC 30107 standard definition of a "presentation attack"—"presentation of an artifact or human characteristic to the biometric capture subsystem in a fashion that could interfere with the intended policy of the biometric system" [14]—to refer to attacks involving falsified biometrics.

Presentation attacks have been a concern in biometric systems since the field's inception. An attacker can utilize knowledge of, for example, a user's fingerprints in order to fabricate a fake finger that he/she can then apply to the sensor and foil the system. Similarly, an attacker can hold up a photograph of the user's face before the camera in an attempt to unlock a mobile device that uses face recognition as a gatekeeper. Biometric researchers, manufacturers, and standardizing groups (e.g., International Organization for Standardization Standards Office and National Institute of Standards and Technology [NIST]) are currently working to develop efficient methodologies to stop such attacks. With the increasing reliance on mobile biometrics in government applications, NIST developed a protocol to ensure security in mobile device biometric applications [15]. Selfie biometric systems in mobile devices have added a sense of urgency to these efforts; although proven vulnerable, millions of consumers and organizations continue to rely on the security afforded by face and fingerprint recognition on their mobile devices.

Protecting devices against presentation attacks is challenging. First, additional hardware may be needed to allow biometric sensors to differentiate between a real biometric and a spoof. The increased costs and design complexity are problematic for mobile selfie biometric systems wherein strict size and cost constraints pose issues. Second, many mobile devices have limited computational resources, which precludes the use of the best available software approaches that can be computationally intensive.

Next, we discuss presentation attacks and countermeasures for face, fingerprint, and voice modalities that are commonly used in mobile device biometrics.

16.3.2 Face Presentation Attacks

Face presentation attacks are classified based on whether they use 2-D or 3-D face artifacts. 2-D attacks typically involve the use of 2-D face videos, still photographs, and other forms of 2-D artifacts to deceive systems that identify people based on 2-D images (as many mobile selfie biometric systems do). 3-D attacks use 3-D masks and other types of 3-D artifacts to deceive face recognition systems based on 2-D or 3-D face images. According to a survey conducted by Galbally et al. [16] and Rattani and Derakhshani [17], the specific techniques in these categories can be summarized in the following ways.

(1) Photograph attacks: These involve presenting the camera with a 2-D face photograph of the legitimate user. The image can be printed on paper or displayed on the computer screen. This type of attack was successful against early versions of Android's face unlock feature [18] as well as modern Android-based devices (e.g., Samsung Galaxy S8 [18]).

If the system requires that the user blink, an animated image that mimics blinking can be created. Another technique involves creating a copy of the original face image, creating another image with the eyes erased, and then rapidly alternating these two images on the computer screen positioned in front of the camera. Such a technique was used to defeat the blinking detection mechanism in face unlock that was introduced in Android to counter presentation attacks [18].

Blinking can also be faked using a printed 2-D mask of the face with holes cut out for the eyes and mouth. The attacker then wears the mask in front of the camera and replicates natural blinking and mouth movements as needed [19].

(2) Video attacks: The attacker presents the camera with a video of the legitimate user's face. The video preserves a face's movements and texture and therefore can defeat rudimentary anti-spoofing mechanisms such as blinking [20]. This attack has been successful against mobile banking applications that utilize face recognition [19].

(3) 3-D mask attacks: Here, the attacker uses a 3-D mask of the legitimate user's face. Although the task of creating a 3-D mask is generally more difficult than finding a photograph or video of the legitimate user's face, the task is becoming easier due to the availability of 3-D printers capable of cheaply producing high-quality masks and services; for example, www.thatsmyface.com, for a current fee of $299 (at this point in time), can create a 3-D wearable mask from a 2-D face photograph. Another variation of a 3-D mask attack was used to bypass iPhone X's face recognition system based on 3-D imaging [21].

The research into 3-D mask attacks and defenses has recently accelerated due to the availability of datasets featuring different types of 3-D masks, such as 3-MAD [22].

Galbally et al. also discuss *feature-level dynamic*, *feature-level static*, *sensor-level*, and *score-level* approaches for defeating presentation attacks.

Feature-level dynamic approaches analyze the movements of the different face regions in order to detect a still 2-D photograph. The central idea is that the movements of the real face and the movements of the printed image will be different. They

can also use challenge–response protocols requiring that users blink or make specific face gestures, such as smiling or turning. Although feature-level static techniques can help defeat attacks based on 2-D still photographs, they are less effective against video spoofs that contain natural movements. They do, however, make video spoofing attacks somewhat more difficult, as the attacker must find or fabricate a video of the victim performing a specific gesture.

More advanced feature-level countermeasures include comparing the movements of the foreground and background, implementing techniques that use local binary patterns (LBPs) [23] in order to track face movements or detect texture properties of a live face, analyzing face photographs taken in sequence in order to infer the 3-D structure of the face, and estimating the noise resulting from capturing the photograph.

Although Galbally et al. argue that face anti-spoofing, feature-level dynamic techniques require multiple face images during authentication and hence will not work in applications wherein a sequence of face photographs is unavailable, we believe this will not be a problem in the majority of mobile selfie biometric systems wherein such sequences can be readily captured from the device camera. However, if only one image is available, then the feature-level static approaches can be applied; these are generally faster yet tend to be less robust than their feature-level dynamic counterparts.

Feature-level static analysis techniques detect spoofed face images based on the single image rather than a sequence of images. Many techniques in this category are based on analyzing the texture of the face [24]. If a sequence of photographs or a video is available, then these techniques can be applied to individual photographs or video frames. The results of the analysis of each frame can then be fused together and the decision can be made based on the final score. This, however, is believed to be a less robust method than using the feature-level dynamic techniques described above.

Sensor-level techniques differ significantly from static and dynamic feature-level fusion techniques and typically require the integration of additional hardware into the sensor. These can include, for example, an extension of the sensing capabilities with the addition of the infrared or near-infrared (IR/NIR) cameras that capture information beyond the visible spectrum. Recent mobile devices such as iPhone X have recently started using special cameras that construct 3-D face models for face recognition [25], which help defeat spoofing using 2-D photographs and videos. The iPhone X Face ID camera projects IR rays onto 30,000 points of the face to construct a 3-D image of the face. A 2-D IR scan is also captured. According to an Apple white paper discussing Face ID [26]. "This data is used to create a sequence of 2-D images and depth maps" that are then used for authentication.

Although the Face ID system has already been bypassed, it thus far (at this point in time) appears to be more difficult to spoof [26] according to the many documented reports of failed spoofing attempts (i.e., 2-D photographs and videos, 3-D masks) that have worked against other devices. We, therefore, believe that combining IR and 3-D imaging techniques will certainly bring greater security to face recognition on mobile devices. The challenges of doing so require addressing the open research

questions of optimally combining IR/NIR and 3-D data while coping with physical space, manufacturing costs, and computational constraints.

Score-level approaches employ different anti-spoofing strategies. Each strategy is implemented as a module that outputs a score indicating the likelihood that the face image is a spoof, and the scores from the different modules are then combined. The resulting score is then used to judge whether or not the image is a spoof.

Overall, we believe the most effective approaches will occur from combining static feature, dynamic feature, score-level, and sensor-level techniques; each technique possesses unique strengths. Advancements in mobile sensors and computing technologies are also expected to pave the way toward the development of new approaches to combat spoofing attacks. Furthermore, increased computing power capable of scaling increased computational loads imposed by the use of multiple presentation attack detection techniques will make the simultaneous implementation of multiple and simultaneous feature-level static, feature-level dynamic, sensor-level, and score-level techniques a possibility.

16.3.3 Fingerprint Presentation Attacks

Fingerprints are the most popular biometric [27]. Unlike, for example, the face or voice, fingerprints work well in poorly lit and noisy environments. At the same time, fingerprint recognition systems continue succumbing to presentation attacks. Some attacks are as simple as using various sticky materials to pick up a latent fingerprint from surfaces and then apply the captured print to the reader, while more sophisticated attacks include the use of 3-D printed fingers [28].

Marasco and Ross [29] published a survey documenting presentation attacks and proposed countermeasures. We use the survey to guide our discussion of the different types of attacks and countermeasures and then include remarks on their applicability in the mobile device context while discussing and analyzing modern works that specifically focus on mobile devices.

The attack types are categorized based on the methods used for faking a fingerprint:

Cooperative duplication: This occurs when the subject voluntarily presses his/her fingerprint into plaster or a similar material that captures the inverted impression of the fingerprint. The mold is then filled with some liquid material that later hardens and thus captures the actual impression of the fingerprint (e.g., gelatin).

This type of attack can be difficult to execute with mobile devices, as many users are unlikely to cooperate with the process. Indeed, when the authors of this chapter were constructing a multimodal biometric dataset constituting the face, ear, and fingerprints, nearly fifty volunteers were willing to donate their faces and ears. At the point in time this study was written, only a handful were willing to donate fingerprints.

Non-cooperative attacks: These types of attacks do not require the subject's cooperation and are a serious threat to mobile device fingerprint recognition. These can be divided into four sub-categories:

1. **Latent fingerprints**: When a finger touches certain surfaces (e.g., glass, metal, wood), it leaves a fingerprint impression. These impressions may then be collected and used for presentation attacks. Various techniques for collecting fingerprints have been developed [30] and are applicable for mobile fingerprint readers.
2. **Fingerprint re-activation**: When the finger contacts the sensor, it leaves a fingerprint. That fingerprint can be reactivated using techniques such as breathing on the sensor or applying graphite powder.

Earlier generations of fingerprint scanners, such as those used for the Samsung Galaxy S5 [31], required that the user swipe the finger across the sensor. Newer sensors, such as those used for the Samsung Galaxy S9, allow the user to press the finger onto the sensor and hold it in place. This is believed to be more user-friendly than swiping. However, since the swiping motion tends to wipe or at least distort the latent fingerprints (i.e., fingerprints left on the sensor surface from previous contact)—unlike pressing and holding—such an attack becomes a theoretically greater concern. More research is needed in order to establish the real extent of the threat.

3. **Cadaver**: This involves the use of a dead finger to unlock a device. According to multiple reports from law enforcement professionals, it is not uncommon for crime investigators to apply the fingers of corpses to the iPhone fingerprint reader in order to unlock the deceased person's device [32, 33]. These reports come in spite of the claims that anti-spoofing measures in the iPhone fingerprint sensors can successfully discriminate between a living and a dead finger [34].
4. **Fingerprint synthesis** is the use of the user's biometric template stored by the system in order to reconstruct the fingerprint. Such an attack inevitably requires access to the template. Apple and various Android-based mobile device manufacturers currently possess dedicated hardware and software that prevent the compromise of the template data that can frustrate this attack. The details of template security approaches will be discussed in the forthcoming sections.
5. **Other techniques**: Attackers can employ schemes to steal people's fingerprints—e.g., by leaving materials on surfaces often touched by people that capture fingerprints. Materials can include gel, plaster, or forensic fingerprint powders. The attacker can then later return to collect fingerprints.

A more sophisticated form of attack would involve secretly embedding a fingerprint scanner that produces high-resolution images of fingerprints or a device that captures the fingerprint topology in ATM machines and other places that frequently come into contact with human fingers. Indeed, a malicious mobile device manufacturer can choose to purposely leak fingerprint images from the user's phone back to the manufacturer, where they can then be used for presentation attacks.

A person's fingerprints can also be obtained through coercion or secretly without consent—e.g., pressing the finger into gel or plaster while the victim is asleep or

distracted. The efforts and risks may well be worth the reward depending on the attacker's purpose.

Other potential, less orthodox threat vectors may exist as well. For example, it has been discovered that iPhone's Touch ID system allows the enrollment of pawprints of cats [35], dogs [36], and hedgehogs [37]. Some pet owners have allegedly used this technique to protect their phones (although we could not verify the validity of these accounts). Therefore, an attacker with access to the user's pet can replicate the pawprint using cooperative duplication techniques described above.

Next, we discuss techniques for countering the fingerprint presentation attacks. First, we provide an overview of the different types of defense measures and then discuss defense measures used with modern mobile devices.

Marsasco et al. [29] separated anti-spoofing techniques into two categories: hardware-based and software-based. Hardware-based measures require that additional anti-spoofing hardware be integrated into the sensor such that the sensor may discriminate between a live finger and a spoof. Software-based techniques process the biometric data and features in order to detect anomalies that can signal a spoofing attack. Such techniques can be broken down into dynamic and static techniques. Dynamic techniques include detecting ridge-based distortion and fingerprint perspiration properties. Static techniques include detecting anomalies in the finger texture, detecting the pattern of the sweat pores on the finger, and detecting the fingerprint's perspiration properties (using methods different than dynamic). Next, we discuss these techniques in the context of mobile biometrics.

Hardware-based techniques: The integration of hardware-based measures into mobile devices can be challenging and is subject to cost and physical space constraints. We briefly examine some of these technologies developed by Apple and manufacturers of the various Android-based devices.

Optical fingerprint scanners are the oldest method of capturing and comparing fingerprints that rely on capturing an optical image and using algorithms to detect a user's biometric patterns, such as ridges and valleys (see Fig. 16.2), by analyzing the lightest and darkest areas of the optical image. The major drawback of optical scanners is that they are not difficult to bypass, since only 2-D pictures are captured

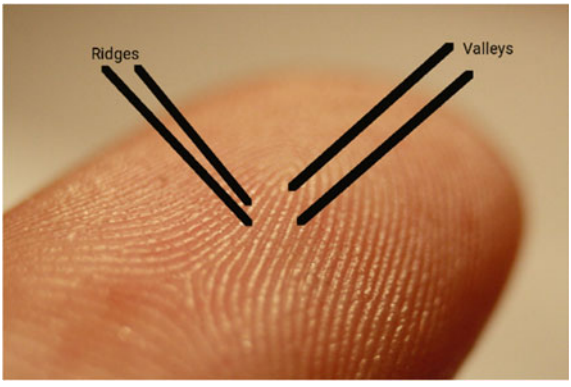

Fig. 16.2 Fingerprint ridges and valleys

and can be replaced with prosthetics or other high-quality pictures. Therefore, this technique is not widely used in modern devices.

The most commonly found type of modern fingerprint scanner is the capacitive scanner. Such a scanner was used in iPhone 5s, which was the first mobile device produced by Apple to support fingerprint recognition. The fingerprint reader used a capacitive sensor to read the pattern of fingerprint ridges and valleys (see Fig. 16.2). Rather than creating an optical image of a fingerprint, capacitive fingerprint scanners use arrays of tiny capacitor circuits to collect data from a user's fingerprint. The advantage of such sensors compared to traditional optical sensors, which simply take photographs of the ridges and valleys, is that capacitive sensors actually require that the finger applied to the sensor has the proper shape. Therefore, such sensors cannot be deceived by simple attacks wherein the attacker applies a fingerprint image to the sensor. Capacitors store electrical charges that are connected to conductive plates on the surface of the scanner to track a fingerprint's details. The charge stored in the capacitor will be changed slightly when a finger's ridge is placed over the conductive plates, and the air gaps between ridges will leave the charge at the capacitor unchanged. An op-amp integrator circuit is used to track these changes by causing the output to respond to changes in the input voltage over time. The result is then recorded by an analog-to-digital converter.

The latest fingerprint technology is an ultrasonic sensor, and Qualcomm's [38] Sense ID [39] ultrasonic fingerprint sensing technology is a major player in this arena. In order to capture the biometric details of a fingerprint, the hardware is composed of both an ultrasonic transmitter and a receiver. An ultrasonic transmitter transmits a pulse against the finger that is placed over the scanner. Some pulses are absorbed, while others are bounced back toward the receiver—depending on the type of biometric traits, such as ridges, valleys, and pores. Hence, depending on the signals received, a map of the fingerprint features is created. These types of scanners require that the fingerprint to have proper shape and hence cannot be deceived with a simple fingerprint photograph.

In order to prevent fingerprint spoofing, anti-spoofing technology can be implemented in software, hardware, or both. Hardware-based solutions have the advantage of a greater ability to detect the liveness of the finger that is scanned, but require additional hardware capabilities in the fingerprint scanner—such as the ability to sense pulse, temperature, and capacitance—that cannot be performed using software alone.

Typical fingerprint anti-spoofing systems measure parameters such as temperature, electrical conductivity, pulse oximetry, and skin resistance, and the built-in logic ensures the sensed value is within an acceptable range. The system includes a fingerprint sensor to capture fingerprint image data, coupled with a spoof detection module that may consist of the following components:

1. Logic that is programmed to determine the probability of a spoof from a combination of metrics derived from the fingerprint image data.
2. A metric generator is included to generate the metrics, and classifier logic is included to generate the raw probability from the metrics that the fingerprint image data was generated from a synthetic material.

3. Adjustor logic is included to adjust the raw probability by a base probability to generate the spoof probability. The base probability is generated from stored metrics based on fingerprint image data captured during an enrollment step.
4. A filter is included to divide the fingerprint image data into multiple windows. The classifier logic is also programmed to determine the spoof probability based on a comparison of the values computed from each of the windows.
5. An access module is coupled to a host system that is programmed to grant access to the host system when the spoof probability is within a predetermined range. This host system can be any one of various electronic devices, such as a smartphone, touchpad, digital camera, personal computer.
6. A storage is also included to store the metrics that are obtained in the previous steps. The storage is coupled to the spoof detection module over a network, and the stored metrics are encrypted.
7. A metric calculator is included and coupled to the classifier logic. The metric calculator is programmed to calculate multiple metrics from fingerprint image data.

Software-based techniques: Next, we discuss the dynamic and static software-based techniques that analyze sequences of fingerprint images to detect spoofs. Image sequences can be captured while the user holds the finger to the sensor for a few seconds. Marasco and Ross documented the following dynamic techniques:

1. **Perspiration-based techniques**: It is common for fingers to perspire. These approaches analyze a sequence of fingerprint images captured over a short period of time to track the progressive flow of sweat that originates in the sweat pores (located along the fingerprint ridges) and moves across those ridges. The presence of sweat makes the ridge areas between pores appear darker than the surrounding areas. The presence of these patterns is evidence of a live finger. To the best of our knowledge, this approach has not yet been attempted on mobile devices although presents an interesting research opportunity.
2. **Ridge distortion techniques**: When the finger is moved around the sensor while being pressed, the resulting fingerprint image becomes distorted. Unique properties of the skin produce significantly greater distortion for a live finger than a spoofed one. The amount of distortion can be measured by assuming the first image is non-distorted and then comparing the distortion in the first image to the other images in the sequence. Specifically, the system can look for a positive correlation between the increase of the fingerprint area and the intensity of the signal, both of which occur when pressure is applied to the surface of the finger.

Static features: Techniques in the category rely on a single image rather than a sequence, which makes the approach more efficient albeit less robust. Static features include the unique texture of the skin, properties of the skin elasticity, or perspiration-based features.

Live and spoofed fingerprints have different textures characterized by morphology, smoothness, and orientation. Marasco and Ross identified the following texture-based approaches:

1. **Texture-based**: Materials such as silicon and gelatin that are commonly used for creating spoofed fingerprints tend to be less smooth than the skin of a real finger. The extra coarseness can be measured in terms of the standard deviation of the remaining residual noise after the original image is denoised, wherein a larger deviation would be associated with a coarser surface. This approach, as is the case with most static approaches, is expected to scale favorably to mobile devices. Avila et al. [40] seem to agree, and they discussed this technique in their technical report on state-of-the-art liveness detection measures for mobile devices.
2. **First- and second-order statistics**: A live fingerprint can be distinguished from the spoof based on the differences between probabilities in observing a particular gray value at the random location on the image or based on the non-uniformity of gray areas distributed along the ridges due to sweat pores and other factors in the fingerprint anatomy. First-order statistics (i.e., mean, energy, entropy, median, variance, skewness, kurtosis, and coefficient of variation) can be used to model the distribution of gray levels, while second-order statistics construct the joint gray-level function between pairs of pixels. Both types of statistics can be efficiently computed on a modern mobile device.
3. **Local-ridge frequency analysis**: This approach [41] is based on multi-resolution texture analysis and inter-ridge frequency analysis. It measures how the distribution of the gray levels in the fingerprint image changes in response to the changes in the fingerprint structure. Moreover, *cluster shade* and *cluster prominence* features are used, both of which are computed based on the co-occurrence matrix constituting the joint probability function of two elements in a given direction and distance. Finally, these multi-resolution analysis features are combined with ridge frequency features, and a fuzzy-C-means classifier is then used to classify the combined feature set as legitimate or illegitimate.

As Marasco and Ross point out, this approach benefits by *not* depending on the perspiration phenomenon. However, local-ridge frequency analysis can be affected by cold weather, skin conditions, and dirt and moisture on the finger. This can be very problematic in mobile use cases wherein fingerprints are expected to operate in uncontrolled conditions that often include the aforementioned situations.

4. **Local phase quantization (LPQ) analysis**: A fingerprint can be rotated in many different ways. A rotation invariant LPQ technique can identify the spectral differences between a legitimate fingerprint and a spoof. The technique has the advantage of remaining robust against blurring and is likely to scale well to the mobile device's limited resources, as evidenced in the work by Jiao and Deng [42], who used LPQ in an indoor positioning application based on the mobile device camera.
5. **Power spectrum analysis** [43]: Creating a spoofed fingerprint changes the frequency details between the ridges and valleys of a fingerprint. This, in turn, results in a spoofed fingerprint image containing fewer high-frequency characteristics than a live fingerprint. The amount of high-frequency data can be computed using Fourier transform. There are currently many libraries, such as TarsosDSP [44],

that support Fourier transform. Therefore, we believe the technique is likely to prove viable for mobile devices.

6. **Local binary patterns (LBPs)**: Real and spoofed fingerprint images have different textural characteristics that can be described using LBP features. The original LBP algorithm was proposed by Ojala et al. [45] and assumes a localized 3×3 pixel image region. LBP is then computed by subtracting the gray value of the center pixel from the other pixels in the region. If the difference is less than or equal to 0, then the result is 0; otherwise, it is 1. Thus, the values of the surrounding pixels are binarized. Finally, the binarized value of each pixel is multiplied by the original value, and the results are summed in order to obtain the LBP operator. Mobile device libraries, such as OpenCV [46], include functions for extracting LBP.

Nikam and Agarwal [47] fused LBP features with wavelet-based features to represent ridge frequency and orientation information. The dimensionality of the fused dataset was then reduced using the Sequential Forward Floating Selection (SFFS) algorithm [21] and was then classified using a hybrid classifier approach that combined neural networks, a support vector machine, and the *k*-nearest neighbor (*k*-NN).

Jia et al. [48] argued that the 3×3 area fails to capture some useful textural information of the fingerprint. To address this, they proposed a multi-scale LBP operator (MSLBP). In their work, they utilized two approaches: (1) increasing the radius of the area beyond a single pixel, and then (2) applying filters to the original image as well as applying an LBP operator in the fixed radius. Evaluation on the Liveness Detection Competition 2011 (LivDet2011) database [49] showed a significantly greater increase in spoofing detection accuracy compared to the traditional LBP approach.

A more recent work by Kumpituck et al. [50] proposed that LBP be used to characterize the local appearance of sub-band images— images coded using the sub-band coding technique that decomposed the image into different constituent frequencies and then encoded each frequency separately. They first decompose the original image using a two-dimensional discrete wavelet transform (2D-DWT) in order to obtain a sub-band image. LBP extraction is then performed on the resulting image, and the extracted features are used to train the SVM classifier. The trained classifier is then used to classify images as either live or spoofed. The authors then evaluated their approach using the LivDet [49] database containing spoofed fingerprint samples and reported that using LBP derived from the sub-band images more significantly improves spoofing classification accuracy compared to the traditional approaches that use wavelet energy from sub-band energy. 2-DWT [51] as well as LBP extraction has been previously performed on mobile devices and is thus computationally viable.

Other approaches: Other static approaches include Weber Local Descriptor [52] and Binarized Statistical Image Features [53], both of whose computational demands consist primarily of linear algebra operations that can be efficiently implemented on modern mobile devices.

Perspiration-based features:

1. **Individual pore spacing**: Perspiration around the perspiration pores results in a recognizable pattern of gray levels. FFT can be used to detect these patterns. FFT is currently well supported through existing developer libraries for mobile devices [54].
2. **Intensity-based features**: Works in this category attempt to distinguish between real and spoofed images based on the uniformity of gray pixel distribution. Researchers have observed that live fingerprints have a non-uniform distribution of gray levels as well as high ridge/valley contrast values. In addition, depending on the material used to create the spoofed fingerprint, the spoofed fingerprint images have been observed as exhibiting less variation in the gray levels.

The conversion of gray images into grayscale and the analysis of pixel values comprise a computationally non-intense process performed routinely in image processing applications implemented on mobile devices.

Quality-based features: These approaches focus on discriminating between live and spoofed fingerprints based on image quality. Quality differences can be measured in terms of strength, continuity, and clarity of ridges. The hypothesis here is that spoofed images will be weaker, less continuous, and exhibit fewer clear ridges. The continuity can be measured by considering the energy concentrations, which can be computed using basic statistical and linear algebra techniques. For example, the ridge strength can be computed as a ratio of eigenvalues of the covariance matrix and the gradient vector. Similarly, the ridges' clarity can be computed using the mean and standard deviation of the foreground image [52]. The relatively low overhead of such computations makes them well suited for implementation on mobile devices.

Furthermore, as Marasco and Ross pointed out, pores located along the ridges are difficult to spoof. Therefore, integrating quality-based features into mobile device systems may be a promising approach to add yet another obstacle for frustrating fingerprint spoofing.

Pore-based approaches: Manivan et al. [55, 56] firstly used a high-pass filter to identify active sweat pores and secondly used a correlation filter to determine their position. Others have experimented with techniques to analyze the number [57] and distribution of pores [58] as well as the detection of active pores [59, 60] on the fingerprint image, the main hypothesis being that differences exist between live and spoofed images.

Rattani et al. suggested that the existing software-based anti-spoofing fingerprint methods are not robust across fingerprint fabrication materials [61]. The performance significantly drops when the fingerprint—fabricated using novel materials—is classified during the testing stage. To mitigate the impact of novel fabrication materials, automatic adaptation [62], image preprocessing [63], and open-set, classification-based [62] anti-spoofing schemes are proposed.

The above techniques should be computationally scalable to modern mobile devices. Multiple packages that support efficient implementation of low- and high-pass filtering techniques are currently available for Android [64, 65]. The remaining statistical techniques used in these approaches may either be implemented from

scratch or leverage the utilities provided by existing libraries, such as TensorFlow [66]. In terms of effectiveness, more evaluation is needed for images obtained from mobile device fingerprint readers. However, similar to previous techniques, we believe the integration of these techniques provides a promising means to add yet another obstacle to frustrating spoofing attacks.

16.3.4 Voice Presentation Attacks

Voice is an appealing modality for use with mobile devices, as it allows users to interact with the device naturally through speech—the most common means for human communication. It is currently being used with Android and iPhone devices to interact with digital agent programs, thus allowing the user to perform tasks on the device by iterating commands. In addition, Android's voice unlock feature allows users to unlock their devices by uttering "Ok Google." The feature recognizes users based on their unique voice characteristics.

However, the use of voice recognition for secure authentication on mobile devices remains limited. We believe this is a result of the difficulty associated with the threat of voice spoofing attacks. Indeed, Google's support warns users of voice unlock: "You can let 'Ok Google' unlock your device when your Google Assistant recognizes your voice. Note: This setting can make your device less secure. A similar voice or recording of your own voice could unlock your device" [67]. This warning refers to the well-documented threat of a replay attack where the impostor records the legitimate user's voice and then replays it. Young et al. [68] have analyzed replay attack vulnerabilities in mobile device voice recognition systems and have proposed replay attack methodologies that can be performed using easily available software and hardware (e.g., the Raspberry Pi computing device [69]). They built a device that connects to the victim's phone and injects commands to the phone's digital assistant. Google's warning also refers to attacks involving zero-effort impostors, wherein the impostor simply speaks in his/her original voice hoping the system will mistake his/her voice for that of the legitimate user, or more sophisticated attacks wherein the impostor uses electronically synthesized speech or attempts to speak in a way that mimics the speech of the target user (such an impostor may potentially require significant training and experience).

Voice recognition can either be text dependent, wherein the same phrases must be used during the enrollment and authentication stages, or text independent, wherein any phrase can be uttered during authentication and the recognition is based on the sound of the user's voice. Text-dependent recognition is generally associated with achieving greater recognition accuracy with shorter phrases [70] and hence proves more convenient for mobile devices than text-independent speech recognition. Both types of systems, however, would face the challenge of dealing with spoofing attacks. The effort involved in simply recording and replaying a voice can be as simple as using an application on another mobile device to record a legitimate user while he/she attempts to unlock his/her device. Therefore, any user with a mobile device can be

a potential attacker, which includes 77% of Americans as of 2018 [71]. We discuss potential solutions from traditional, non-mobile voice recognition systems that may prove useful in the context of mobile biometrics.

Spoofing vulnerabilities: Commonly used voice features include short-term spectral, prosodic, or high-level features. Short-term spectral features are derived from short voice frames (e.g., 20–30 ms long) and are used to describe voice timbre. Commonly used short-term spectral features include mel-frequency cepstral coefficients (MFCCs), linear predictive cepstral coefficients (LPCCs), and perceptual prediction (PLP) features [70].

Prosodic features are syllables and words that describe speaking style and intonation. The use of prosodic features for authentication may not be ideal with mobile devices because they require relatively significant training data, which might be inconvenient for the user to supply. In addition, prosodic features based on pitch are not robust in uncontrolled conditions [70] in which mobile devices operate.

High-level features include word usage, pronunciation, and other types of information that can be parsed from discrete tokens of speech. These can be robust to environmental noise but may require preprocessing in order to convert speech to text from which high-level feature extraction is possible.

All three types of features can be spoofed. Short-term spectral features can be spoofed by simply recording and replaying speech. Modern voice synthesizers are also capable of reproducing short-term spectral features if given the model of the speaker's voice.

Prosodic features can also be reproduced using synthesizers and voice conversation systems. One approach is to use a voice synthesizer to generate fundamental frequency trajectories that are correlated with the voice of the speaker being impersonated [70].

High-level features are based on speech content and can thus be easily spoofed by replaying the speech, which will have the same spoken phrases as the original voice. Moreover, artificial intelligence systems and statistical models can be used to generate speech with content sufficiently similar to that of the impersonated speaker. Next, we discuss specific threats and their countermeasures. Specifically, we discuss countermeasures to attacks based on recorded and replayed speech, synthetic speech, and voice conversion and impersonation. Our discussion is guided by the survey published by Wu et al. [70] although relates the attacks and countermeasures to mobile use cases and presents discussions of modern publications specifically targeted toward mobile devices.

Record and replay attack countermeasures: The original approach to detecting recorded and replayed speech was proposed by Shang and Stevenson [72]. The approach is based on storing voice samples from past authentication attempts and comparing these samples to the access phrase used during the authentication attempt. The attack is considered a replay if the new sample closely matches one of the prior samples. Such a technique may prove impractical for use with mobile devices, as storing all prior samples would likely result in excessive storage space consumption. In addition, the attacker might be able to obtain a sample recording sample of the user (e.g., from an online video) that was not previously used for authentication.

Villalba et al. proposed that the increased noise and reverberation resulting from replaying far-field recordings be used to detect spoofing [73]. Although the technique was effective in significantly reducing the false acceptance rate (FAR), it was attempted on both the landline and GSM telephony systems yet not on mobile devices. The approach's effectiveness for modern mobile devices remains unclear because much depends on the microphone and speaker technologies used in the attacks; these can also vary widely across devices.

Wang et al. used channel noise to detect voice samples that were recorded and replayed. The hypothesis was that the voice sample originally recorded from a live human being would only contain channel noise from the device used by the voice recognition system [70, 74]. A sample obtained from a replayed recording would also contain channel noise from the recording device and the speakers used for replay. The approach was effective in reducing equal error rates (EER) from 40.17 to 10.26% when a system based on Gaussian mixture model–universal background model (GMM-UBM) was subject to spoofing attacks. We believe this technique can scale to the limited computational resources of mobile devices, as GMMs have previously been used in mobile speech applications [75]. The technique's effectiveness in practice would require evaluation using a database of voice samples recorded on a mobile device containing spoofed samples.

Synthetic speech attack countermeasures: Many techniques have been proposed for countering attacks involving synthesized speech. These efforts are in good measure considering that the vulnerability of voice recognition systems to voice synthesis attacks is a well-recognized problem (e.g., [67]).

The synthesis processes are known to introduce detectable artifacts. Satoh et al. have used intra-frame differences that were later demonstrated to work well for synthesizers based on hidden Markov models (HMMs) that do not employ global variance compensation [70]. Other artifacts have been observed, such as the smoothing of high-order cepstral coefficients by the HMM training and synthesis processes resulting in synthetic speech containing less variation than speech originating from a living human being [70, 76]; furthermore, some researchers have focused on studying the acoustic differences between natural and synthetic speech as a means of detecting spoofing attempts. Although the above approaches may scale nicely to modern mobile devices, to the best of our knowledge, few works have focused on countering the threat of speech synthesis attacks on mobile devices.

Voice conversion attack countermeasures: While speech synthesis attacks convert text to speech, voice conversion attacks use speech samples from the targeted user to automatically convert an impostor's voice into a voice that sounds similar to that of the target speaker [70]. The conversion process introduces detectable artifacts that include the absence of the natural phase in converted speech [77, 78] and more decreased dynamic variability compared to natural speech [79]. The authors in [79] also demonstrated that supervector-based SVM classifiers can effectively detect voice conversion attacks based on utterance-level and dynamic speech variability [80], while the approaches based on detecting natural speech phases were argued to likely prove ineffective for cases wherein converted speech preserved the natural phase feature [81].

SVM-based machine learning has been widely used with mobile devices, including by the authors, which scaled adequately [82]. Therefore, we believe the implementation and evaluation of the technique in mobile applications is a viable topic for further investigation.

Human-based voice impersonation attack countermeasures: In contrast to the speech synthesis or speech conversion attacks that involve the use of technology to impersonate the voice of another person [70], a human-based voice impersonation attack does not require any additional equipment, but rather simply involves one person attempting to speak in a way that resembles another. The studies evaluating the effectiveness of these types of attacks have reported contradictory findings and are thus inconclusive.

Part of the challenge in developing countermeasures against this type of attack is that human-based voice impersonation involves the use of natural speech and hence often lacks the detectable artifacts resulting from record and replay, voice synthesis, and voice conversion [70]. Nevertheless, Chen et al. [83] successfully used the Spear system developed by Khoury et al. [84] in order to construct a mobile system resilient to human-based spoofing attacks. The system was implemented on Android 4.4 KitKat smartphone and was based on the Gaussian mixture and intersession variability (ISV) techniques. The system yielded low FARs when evaluated by the Carnegie Mellon University (CMU) Arctic Database [85].

Other recent countermeasures: Chen et al. [83] proposed a software-based approach for mobile devices that detect recorded and replayed voices based on the magnetic field emitted by the speakers. The hypothesis is that, unlike humans, loudspeakers use magnetic force to create sound that in turn produces a magnetic field that can be detected using the magnetometer sensor within a mobile device. The authors also used a Spear system [84], as described in the previous section, to detect human impersonation attacks. The overall system was able to achieve 100% accuracy.

Feng et al. [86] developed a small wearable device that protects mobile device digital assistants against replay, speech synthesis, and human-based impersonation attacks. The device includes an accelerometer that is agitated by the speech signal. The accelerometer data is then communicated via Bluetooth to the mobile device, where it is correlated with the sound data received from the mobile device microphone. This correlation is then used to perform matching on the remote server. The system produced a 0.1% false positive (or acceptance) rate. Although the wearable component may present usability concerns for users, it also presents new opportunities. For example, the wearable component can also take on the function of the security token used in multifactor authentication. Matching the voice on the remote server may prompt privacy concerns from users who fear their voices might be recorded and stored on remote systems for espionage purposes.

Zhang et al. [87] developed a mobile-based approach using the Doppler phenomenon to resist replay and human-based spoofing attacks. When the user utters a passphrase, the phone's speaker emits a 20 kHz tone—a high-frequency sound inaudible to the human ear—and monitors the microphone to pick up the signal reflections. Those resulting from the movements of the user's lips, vocal chords, etc.

while uttering a passphrase cause Doppler shifts that are used to evaluate the voice's liveness. During evaluation, the system achieved a 1% error.

Zhang et al. [88] proposed a voice replay attack detection system that leverages mobile devices' stereo sound recording capabilities. The central idea is that a stereo recording system uses two microphones, and when the live user speaks while holding the phone close to his/her mouth, the voice signal arrives at the two microphones at different times. The same phenomenon does not manifest in the case of replayed recordings.

16.3.5 Multimodal Biometrics

Using multiple biometrics requires the user to provide more evidence in order to prove identity and hence increase the amount of identifying data the attacker needs to spoof. However, combining data from multiple biometrics in a way that makes the system resilient to spoofing attacks is challenging.

Rodrigues et al. [89] empirically demonstrated that, in multimodal biometric systems that combine match scores from different modalities (using weighed sum, likelihood ratio, and Bayesian likelihood ratio), bypassing a single modality may suffice to bypass the entire system. Therefore, the multimodality of such a system simply presents the attacker with opportunities for spoofing.

Combining identifying data at the feature level is associated with greater recognition accuracy compared to combining match scores. We have previously developed feature-level fusion schemes for mobile devices that achieved significantly lower errors [82, 90, 91] compared to unimodal schemes in the presence of zero-effort impostors. We also believe feature-level fusion is a more promising approach toward multimodal systems' resilience to spoofing attacks than are methods based on score-level fusion of modalities. We are currently in the process of evaluating the performance of these schemes against spoofing attacks.

Below, we first propose methods and techniques for strengthening multimodal biometric systems on mobile devices against spoofing attacks by dividing the approaches into software- and hardware-based.

Software-based: We believe that, as the first line of defense, a multimodal system on a mobile device must perform spoofing detection on the individual modalities. Ideally, at the software level, the spoofing detection on each modality should be performed using multiple techniques to maximize the probability of detection.

The second line of defense may constitute techniques that exploit the system's multimodality to detect spoofing. For example, within a system based on face and voice, the voice signal can be correlated to the movement of the lips. A system based on the face and ears also presents multiple opportunities for increased spoofing detection. For example, our research group is currently researching the feature-level fusion of face and ear biometrics on a mobile device. The user interacts with the system by looking straight at the camera, which captures the face, and quickly turns his/her head to both the left and right such that the camera may capture both ears.

We believe the properties of these motions can be analyzed to detect replay attacks by, for example, analyzing the subtle sound made by the motion or using the device accelerometer to measure vibrations created by the motion.

The third approach to detect spoofing of a multimodal biometric system involves studying the properties of a fused set of features where one or more modality is being spoofed. We believe a correlation of the fused feature's properties set to modality spoofing may present a new line of promising research. Such research is currently being undertaken by our research group.

Finally, the software-based techniques should also be backed up by sensor-level techniques and other hardware-based techniques that may further increase the resilience of the system.

Hardware-based: To have a multimodal biometric system that is reliable and effective, it is necessary that embedded hardware be employed—e.g., low-power processor, digital signal processor (DSP), or field-programmable gate array (FPGA). These hardware resources can be used in various real-life applications, such as the authentication of electronic identification for driver licenses and e-passports, user authentication within financial institutions, and entry control within buildings, laboratories, and borders. As stated above, multimodal biometric systems can raise security to another level by adopting more than one biometric trait.

Most multimodal biometric systems are required to have powerful computing environments in which complex tasks can be executed at reasonably high speeds. Using software alone, it is not easy to process multiple biometric traits with different features in a reasonable amount of time. Therefore, we need a multimodal biometric system with support from efficient hardware in which various multimodal biometric algorithms are performed on a real-time basis. Typical application processors used in most embedded systems work at a clock rate of only a few hundreds of MHz, and the floating-point arithmetic is not hardware-implemented. However, multimodal biometric algorithms that process multiple biometric traits in parallel require higher computing power with hardware-implemented floating-point arithmetic in order to ensure a real-time authentication.

In order to perform multimodal biometrics in real time, some tasks and executions that require high computational power can be implemented into FPGA. These tasks are dynamically synthesized on FPGA, and the multimodal biometric algorithms can be processed significantly faster. Most multimodal biometric algorithms are directly related to digital image processing because multiple biometric traits, such as faces and fingerprints, are required. In the recent years, embedded system performance has been increased due to the development of new hardware such as low-power processors, DSP chips, and FPGAs. Among the hardware, FPGA is a promising technique to be used in multimodal biometric systems because it may accelerate the execution of algorithms and offer tremendous potential toward improving overall performance through parallelization [92].

Although recent new processors' technology continuously improves the performance of multimodal biometrics, the potential of implementing these algorithms on the CPU is still not fully exploited. An FPGA device can accelerate the execution of algorithms and offer a tremendous throughput by employing parallelization. On the

other hand, for multimodal biometrics, the FPGA cannot accommodate all required algorithms. Therefore, optimization at the software level and hardware implementation at the hardware level must be carefully considered. Herein, the software and hardware co-design is used to design the system, which consists of both a hardware platform and software platform. The hardware employs Intel DE5 board within two DDR3 SODIMM slots that can be used to expand the amount of memory available to the FPGA. The FPGA board connects with CPU through peripheral component interconnect express (PCIe). Consequently, the biometric algorithm (e.g., face fingerprint modules) runs on the hardware platform. Our experimental results reveal that the proposed software and hardware hybrid platform can achieve three times the acceleration that the software counterpart can achieve.

16.4 Template Security

The widespread use of the biometric systems requires massive storage of biometric data. In the generic biometric authentication system, there are five major components: sensor, feature extractor, template database, matcher, and decision module (see Fig. 16.3). In Fig. 16.3, two procedures of the biometric system are depicted. During the enrollment procedure, the user information is stored in the template database. On the other side, the biometric sensor is the interface between the user and the authentication procedure. The function of the biometric sensor is to collect the biometric trait of the user. Then, the quality assessment model determines whether the collected biometric trait is sufficient for further processing. The feature extractor processes the collected biometric data to extract salient information for distinguishing between different users. Once the user information can be found in the template database, the matcher module can execute a program that compares two inputs from

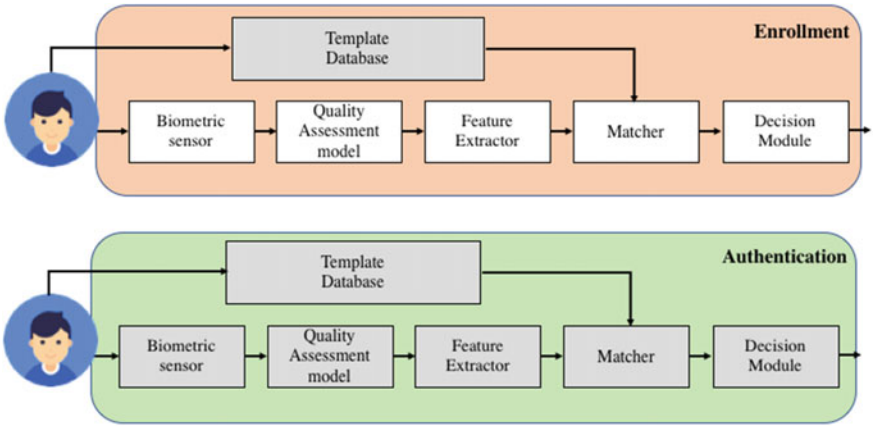

Fig. 16.3 Enrollment and authentication stages in a biometric system

the template database and feature extractor as well as generates the output as a match score. Finally, the decision module makes the decision.

The protection of the template database is not a trivial task, and some works have employed template protection schemes to improve security in a template database [93]. Non-reversibility is introduced to define the computationally infeasibility of recovering the unprotected template from the protected template. Therefore, individuals exploit the possibility of creating different protected templates from the same template used in various applications—a property known as *diversity*. Consequently, diversity leads to *revocability,* which involves protecting as many templates as necessary. Current template protection schemes can be divided into two categories: *feature transformation systems* and *biometric cryptosystems*. Some previous works have been proposed as being inspired by feature transformation. For example, the Biohashing system combines the password provided by the user with biometric data [94]. However, this method requires many passwords to protect templates, and these passwords must be stored privately. On the other hand, biometric cryptosystems try to generate additional information for unprotected templates. The major contribution of this method is that additional data is not required to be kept private. Some works provide insight into possible attacks within the generic biometric system (see Fig. 16.1). Although the software-based solution aims to protect the template database, the delay and security of the protection module are considered major drawbacks.

Some works [95] propose a fingerprint biometric cryptosystem for an FPGA device. The results imply that accuracy is improved and delay is reduced. To implement a fingerprint biometric cryptosystem in the FPGA device, both the algorithm and hardware architecture must be considered carefully. In the algorithm aspect, biometric cryptosystems are based on the fuzzy commitment that constitutes error correction and cryptosystems techniques. The error correction code can be processed by two different types either bit-by-bit or block-by-block. Both types are applied in the biometric cryptosystems with Bose–Chaudhuri–Hocquenghem (BCH) and Reed—Solomon codes [96]. On the other hand, the cryptosystems are designated based on *QFingerMaps*, and the additional cryptosystem information is generated by a fuzzy commitment scheme, which fuses the codeword and *QFingerMap* in an obfuscated way. In this work, we employ an (Exclusive OR) XOR operator to fuse the codeword and *QFingerMap*. The main functional blocks that should be implemented on the FPGA device include *QFingerMap* extraction, the encoder to generate the codeword, the hash function to protect the generated codeword, and the decoder to correct errors.

16.5 Usability

In this section, we discuss the usability issues affecting selfie biometric systems on mobile devices. We firstly discuss the general principles of user-friendly biometric system interfaces for mobile devices, then provide insights specific to designing friendly user interfaces for mobile device multimodal systems that we have learned through our research and practice.

We then present a novel approach for performing multimodal biometrics on an FPGA that can be integrated with mobile devices and can drastically reduce execution time and power consumption in multimodal mobile biometric systems, as our results suggest. Reducing these aspects is important for improving user experience. The prototype multimodal feature-level fusion system used was taken from [82] and was based on combining face and voice biometrics using discriminant correlation analysis (DCA) as well as classification using *k*-nearest neighbors (*k*-NN). The challenge stems from implementing *k*-NN on the FPGA in a way that is viable for mobile devices.

Software-based: Any software–user interface must be designed to maximize the quality of user experience. Hence, the principles of effective interface design apply to mobile biometrics systems. In this section, we focus solely on specific challenges in mobile device authentication systems that we have learned during our research and practice. We then include a specific discussion of multimodal biometrics.

First, the biometric authentication process should be easy to enable and configure, which is especially important for users who are not technology savvy. Although mobile device manufacturers are making great strides in simplifying the process, some users do not use biometric authentication because they are unsure how to set it up (in our experience, some did not even know where to find the setting). One possible solution involves encouraging the user to utilize the device's biometric feature (if it is fit for authentication) both during and following the device setup process. It is important to ensure, however, that these encouragements be both non-intrusive and easily disabled by the user.

Furthermore, the enrollment process must minimize the amount of user effort; this includes limiting the number of training samples, providing feedback on the user's progress, and minimizing processing time. Otherwise, an initial negative experience may cause the user to give up or turn away from the feature.

Second, the biometric authentication process should be easy to invoke. Many modern devices address this by setting the authentication screen as the first image the user sees upon obtaining the device's attention—typically by pressing a button. Many fingerprint-based systems, such as those used for iPhones and Galaxy devices, allow users to authenticate immediately by placing a finger on the sensor and requiring no prior actions in order to invoke the authentication process.

A similar approach is possible with face- and voice-based biometric systems. For example, many smart home systems, such as Amazon Alexa and Google Home, allow users to get the device's attention by uttering a predefined phrase—e.g., a user can utter "Ok Google" in order for Google Home system to begin accepting commands. However, such an approach would require that the mobile device constantly monitor the microphone or camera, which in turn raises issues of privacy, false device unlocks (e.g., the camera accidentally catches the user's face), and increased power consumption. For example, Android-based phones allow users to conduct Google searches and perform other functions on their devices by uttering "Ok Google." According to previous reports, some users feel apprehensive about their devices constantly "listening" to them through the microphones [97].

Third, the interface should be intuitive when interacting with users and providing them with prompt feedback in the event that matching fails. The latter is especially important for mobile devices, which operate in uncontrolled conditions that cause false rejections. For example, if the fingerprint match fails because the fingertip is wet (assuming the system can detect moisture), then the user should be instructed to wipe his/her finger; or, if the face does not match due to insufficient lighting, then the user should be instructed to increase the brightness. We believe these hints will help reduce user frustration.

Fourth, the matching process must occur instantly, and if successful, the user should immediately be taken to the home screen of the device or to the applications he/she was most recently using. Long authentication times will likely lead to frustration, and the same is true of the enrollment process. Because the enrollment process is typically executed only once, unlike the authentication process that is done repeatedly, greater delays may be tolerated here. To maximize speed, developers can leverage the parallel architecture of modern mobile processors, graphics processing units (GPUs), and other specialized biometric technologies discussed in the section concerning hardware techniques.

Usability of mobile device multimodal biometrics: All the above user interface design principles additionally apply to multimodal biometrics. However, multimodal biometric systems require the collection of multiple biometric modalities, thus requiring greater effort from the user. We believe the key here is to minimize user efforts to a level comparable to that of a unimodal system. One possible way to achieve this is to simultaneously capture samples from multiple modalities.

For example, in our previous work, we experimented with developing an interface for a multimodal system based on face and voice (see Fig. 16.4). A user-friendly GUI for simultaneous capture of face and voice on a mobile device. The interface consists of a live stream from the device's front camera with a square drawn around the user's detected face and a volume meter indicating the strength of the voice signal. Additional indicators are provided to indicate the quality of the face (e.g., luminosity) and voice data, (e.g., signal-to-noise ratio). These indicators utilize percentages, wherein higher percentages indicate greater quality. On one hand, we believe these can help the user quickly identify issues in the event that authentication fails. On the other hand, they can potentially confuse the user with the extra data. We plan to explore the user's experienced utility of these indicators in our future research.

The system records a video of the user's face while he/she utters a phrase. The face images and voice are then extracted from the video track and sound track, respectively. The execution time for the authentication process takes a fraction of a second due to efficient algorithms and parallel extraction of both face and voice features—the most time-consuming operations of our algorithm.

We have also experimented with a multimodal biometric system based on the face and ear, finding that the easiest way for the user to capture both modalities is to look into the camera and then twist his/her head firstly to the left and secondly to the right while holding the device in a fixed position. In our informal preliminary experiments, we observed that users were able to capture both modalities within one second. Figure 16.5 presents a diagram illustrating our approach.

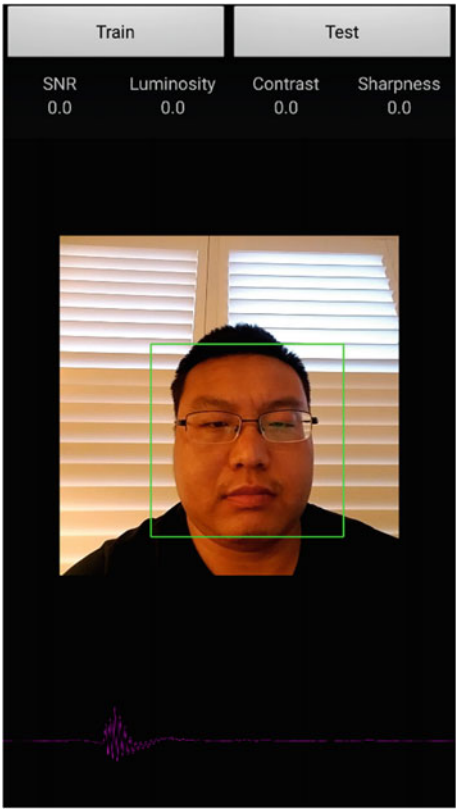

Fig. 16.4 A user-friendly GUI for simultaneous capture of face and voice on a mobile device

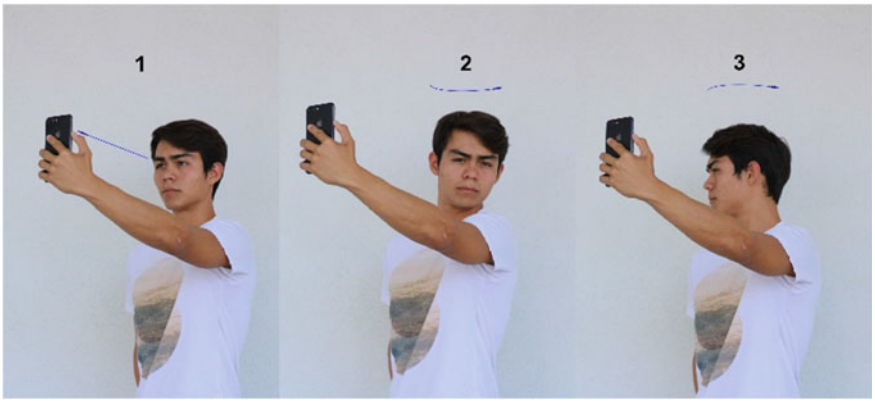

Fig. 16.5 A method for capturing face and ears in a mobile device

The above approach replaces our previous approach, which required that users move the device from the face to the left ear and then to the right ear. However, this proved to be excessively difficult for many users who could not easily find their ears with their cameras.

Samsung Galaxy S9 also includes an Intelligent Scan feature that allows the device to be unlocked with both the face and iris, which are captured simultaneously when the user looks into the camera. We believe this approach is the right direction from the interface perspective.

Expanding authentication to more than two biometrics is even more challenging; however, simultaneous capture can go a long way. For example, the effort required from a tri-modal system based on face, voice, and iris can still be achieved by recording a video of the user's face while he/she utters a phrase and simultaneously capturing images of the iris. Thus, the effort of a tri-modal system is potentially reduced to that of a bimodal system.

Overall, the research on the mobile biometrics user experience is still a relatively new field ripe for future research and innovation. It requires that designers consider technical aspects such as quick execution time, psychological aspects that involve making the system's appearance and operation inviting, and social aspects such as privacy. Achieving this goal will require that software and hardware designers as well as user experience experts join in collaboration to ensure the system is designed bottom up with usability in mind.

Next, we present a novel approach for reducing execution time and power consumption in mobile device multimodal systems using FPGAs.

Fast and power-efficient feature-level fusion of face and voice using FPGA: Current mobile devices can be used to identify users based on a single biometric modality such as the face or fingerprints. However, to attain maximum identification accuracy, prior work has revealed promising results regarding the use of multiple or multimodal biometrics.

Gofman et al. [91] proposed an approach for fusing MFCC features from the face with histogram of oriented gradient (HOG) features on mobile devices. They used DCA to fuse the features and then classified the fused feature set using various classifiers (SVM, *k*-NN, random forests, linear discriminant analysis [LDA], and quadratic discriminant analysis [QDA]). Although their approach led to more significantly improved EER compared to unimodal face and voice approaches, they did not consider the hardware aspects of implementing their approach on mobile devices (e.g., power consumption).

Recently, novel systems were proposed to incorporate programmable hardware into the smartphone to rethink a vision wherein applications may consider both software and hardware components. Current smartphone devices are incredibly constrained energy-wise due to their batteries. Development in battery density has received more attention; however, recent research shows that the battery density has been doubling only every ten years [98]. In addition, the physical size limitations of the portable devices lead to a relatively static energy budget among all devices. Consequently, CPUs empowering modern smartphones are optimized for power effi-

ciency rather than speed. For this reason, implementing the application that requires high computation power on mobile devices nevertheless remains challenging.

k-NN was introduced as supervised and instance-based learning in the early 1950s. This algorithm was not initially popular because it requires high computation power, although it was and remains a popular means of classification in biometrics due to its simplicity and, in many cases, high accuracy. In general, there are three or four steps in the *k*-NN algorithm:

1. Calculate the distance and similarity between the testing set and the training set;
2. Sort the distance and similarity to determine the *k*-nearest classes;
3. Perform majority voting to decide the class.

There are many ways to measure the distance or similarity between data in testing and training sets. The Euclidean, Minkowsky, Chebychev, Camberra, and Manhattan methods for measuring distance are proposed as the following equations [98]:

$$\text{Euclidean: } D(x, y) = \left(\sum_{i=1}^{m}\left(|xi - yi|^2\right)^{1/2}\right)$$

$$\text{Manhattan: } D(x, y) = \sum |xi - yi|$$

$$\text{Minkowsky: } D(x, y) = \left(\sum_{i=1}^{m} |xi - yi|^r\right)^{1/r}$$

$$\text{Chebychev: } D(x, y) = \max_{i=1}^{m} |xi - yi|$$

$$\text{Camberra: } D(x, y) = \sum_{i=1}^{m} \frac{|xi - yi|}{|xi - yi|}$$

In Fig. 16.6, the hierarchical platform-based design for *k*-NN classifier is illustrated. The main purpose of the proposed design is to modularize various functions in both hardware and software. The basic operators including addition, multiplication, square root, subtraction, division, and comparator are depicted. Among various operators, *k*-NN consists of two time-consuming operations: distance computing and

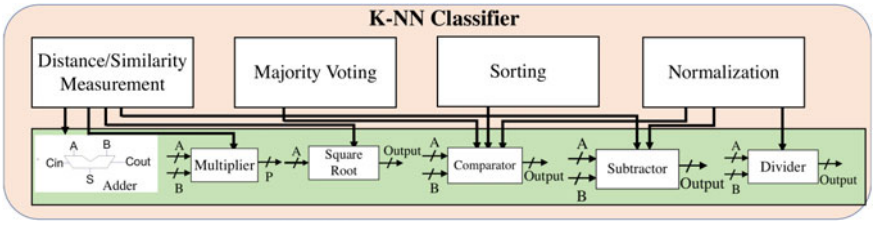

Fig. 16.6 *k*-NN classifier and its hardware functionality

sorting. Thus, these operations can be fully parallelized due to independent distance computation. The parallel property allows us to involve an FPGA device, which is perfectly suitable for implementing such k-NN heterogeneous architecture. In this work, we employ OpenCL architecture to transfer data from the CPU to FPGA. During the distance-computing process, the matrix for distance values is collected between all query and reference objects. Then, the rank process finds the k-NNs for each query object. To sort the distance, the sinking sorting algorithm is employed with a worst-case and average-case complexity $O(n^2)$. The choice of the sinking sorting algorithm in this work was based on the algorithm's property; each candidate is picked up according to the smallest distance in the current queue. The process can be perfectly parallelized because it compares each pair of adjacent items and swaps them if they are in the wrong order.

OpenCL is an open resource framework for parallel programming on systems with heterogeneous processors. Using OpenCL enables multiple hardware architectures by different manufacturers. In Figure 16.7, OpenCL framework connects host processor and FPGA through PCIe connection. The host computer handles the data flow, which is explicitly programmed by the user. The memory system in this work can be categorized into three groups: global, local, and private.

The accelerator is classified into a workgroup sharing the local memory, which plays cache-based memory such that each accelerator can access data stored in the local memory. The global memory is used to store data that is accessible to the workgroup and the host computer, while the private memory is reserved for each

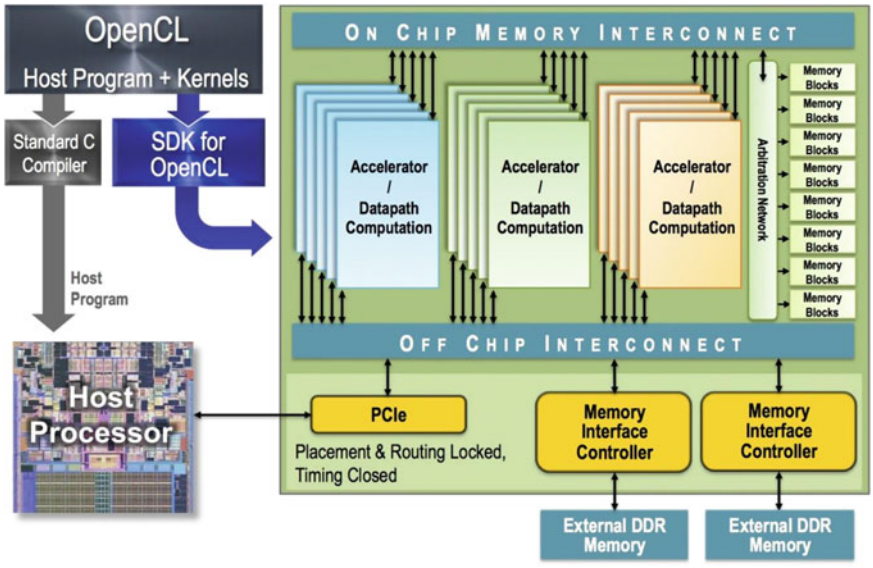

Fig. 16.7 OpenCL compiler to generate both executable software for the system CPU and a bitstream on FPGA [30,000×]

accelerator and performs the fastest data movement. The two functions are implemented in the hardware accelerator.

Compared to a traditional Verilog or VHDL design, the scheduling issue on the hardware resource is automatically attached to the device in OpenCL. Thus, in this work, we need only to design the required number of accelerators and distribute the workload.

Distance Calculation Accelerator

The design of the distance calculation accelerator aims to parallelize the distance calculation at each accelerator. In order to avoid unnecessary latency of memory use, we use local memory for distance calculation. The reference data is loaded into the local memory, which may be easily accessed by the accelerators.

Distance Sorting Accelerator

When a distance calculation accelerator produces the distance matrix between input data and reference data, the distance sorting accelerator is employed to find the k-nearest distance in each row using the sinking sorting algorithm. For example, when the first item compares the third and fourth distances in the row, the second item can be launched to compare others. Once all items have been compared and reached at the end of the row, the k-NNs are formed.

In order to test the approach, we implemented the framework in a CPU and FPGA system. The CPU used in this work was an Intel I7-3770K with 3.5 GHz with a Windows 7, 64-bit operating system. An FPGA board (Intel DE5) with a Stratix V GX was inserted and connected with CPU through PCIe lanes. The integrated transceivers with a transfer speed of 12.5 Gbps allowed the DE5 board to fully comply with version 3.0 of the PCIe standard. Two independent banks of DDR3 SODIMM RAM were used to construct global memory. The local memory employed interconnected on-chip RAM block—a simple process easily given access. Private memory was implemented using flip-flops. The flip-flop within the data flow can run at the accelerator's frequency. To test the framework, we used a labeled face and voice images from [91]. This approach's performance is compared with its CPU counterpart.

Table 16.1 illustrates the performance comparison between the CPU and FPGA results. We utilized twenty different faces and voices in order to avoid some errors during the test. Because the runtime of CPU was longer than FPGA due to unparalleled process architecture, the FPGA could thus achieve 148 times the speed up. Regarding the power aspect, CPU consumed five times that of the FPGA device.

Table 16.1 Performance comparison between CPU and FPGA

Platform	CPU	FPGA
Transistor size/nm	22	40
Runtime/ratio	150.16	1
Speedup/ratio	1	148
Power/ratio	5.41	1

16.6 Selfie Biometrics Case Studies

This section presents a couple case studies on the topic of privacy, confidentiality, and usability of selfie biometrics on mobile devices. The first relates to the face- and fingerprint-based biometric capabilities on the latest smartphones, such as Samsung (Android-based) and iPhone, while the second relates to applications of keystroke dynamics on mobile devices.

Case Study 1

The common modern smartphones mentioned above are equipped with both facial recognition and fingerprint recognition techniques. Android introduced *face unlock* in 2011 [99], while Apple introduced *Touch ID* a couple of years later [100]. These were followed by the *fingerprint lock* on the Samsung Galaxy S7 phones in 2016 and then the more recent *Face ID* technology integrated with iPhone X in 2017 [101]. This was many users' initial exposure to and true interaction with biometrics. It is thus important to assess whether users did or did not choose to adopt any of these biometric-based authentication methods to unlock their mobile devices as well as the underlying reasons behind their decisions. In fact, researchers have stated that the usability of biometric systems is a critical element in users' adoption decisions [102]. Despite the awareness of additional security provided by biometrics through passwords and PINs, people may have concerns about several issues (i.e., privacy and reliability) that act as barriers to the large-scale adoption of this technology among consumers. Hence, studies have been conducted [102–104] to explore users' beliefs, attitudes, and perceptions toward using biometric security on their mobile devices, which remains, nonetheless, not very prevalent today.

Several researchers have performed comparative studies [104, 105] to analyze usability among face recognition, iris recognition, voice recognition, fingerprint recognition, and gesture recognition techniques on mobile devices, all of which yielded considerably critical flaws. In 2014, two studies investigating smartphone unlocking behavior among users had determined that users failed to realize the importance of protecting the data stored on their phones (and hence the risks associated with losing that data) and that users spent more time than necessary to unlock their phones [106, 107]. *Face ID* on the iPhone X has recently become available and has replaced the fingerprint unlocking scheme (*Touch ID*). Reports [108] have mentioned that, although *Face ID* is perceived as more secure than *Touch ID*, there have been several issues with its operation. For example, the former does not work in landscape orientation, it does not always work in bright sunlight or with sunglasses, and it is occasionally slow.

Recently, Bhagavatula et al. [102] explored within-subject usability of *Touch ID* on iPhones and *face unlock* on Android devices in a laboratory setting in order to assess different scenarios in which mobile phones operate. Moreover, they also administered an online survey to 198 participants to evaluate general user perceptions and attitudes toward using different types of biometric security on mobile platforms during everyday life. This study was the first of its kind (at the time) to examine the usability of biometric security on commonly used smartphones in today's society.

We summarize the methodologies used and results obtained from this case study in the following section.

Laboratory usability study: The within-subject study performed by Bhagavatula et al. [102] consisted of comparing four unlocking mechanisms on smartphones: Android face unlock on a Samsung Galaxy S4 phone, iPhone *Touch ID* (fingerprint recognition), Android PIN unlock, and iPhone PIN unlock. They compared these biometric-based authentication schemes because these were the only such schemes available on smartphones at the time (Android fingerprint unlock and iPhone *Face ID* were not yet introduced). The PINs were used as a baseline for comparison among the biometric security techniques. Ten participants—eight male and two females—participated in the study, and each participant was provided with a phone. Participants also filled out a questionnaire concerning their demographic backgrounds, prior experience with smartphones and biometric systems, and perceptions and attitudes toward biometrics (Likert-scale-type questions). Each subject also performed authentication using each of the four schemes in five different scenarios: (1) sitting, (2) sitting in a dark room, (3) walking, (4) walking while carrying a bag in one hand, and (5) sitting and applying moisturizer to the hands. These five situations are consistent with prior studies of mobile phones' user usability. Their main results include

- All participants determined Android face unlock and iPhone *Touch ID* as being easy to use during several common usage scenarios;
- The face unlock did not work for any participants in the darkroom setting;
- *Touch ID* was relatively easy to use even in the presence of moisturizer on participants' hands; and
- Most participants favored iPhone's *Touch ID* over Android's face unlock and PINs.

Online survey: The purpose of the online survey was to understand real usability issues faced by consumers in the real world, such as the perceived usefulness of the biometric security schemes to protect the phone from unauthorized use and the system's ease of use or convenience. For this purpose, 198 subjects who owned a smartphone model that supported either Android face unlock or iPhone *Touch ID* were selected. Similar to the laboratory study, participants were asked through a survey to provide their demographic information, level of prior familiarity with biometric authentication techniques, general phone unlocking behaviors, and rationale for adopting or not adopting a biometric scheme for their mobile phones. The main findings from this survey include:

- Participants using iPhones overwhelmingly perceived *Touch ID* as more convenient to use than PINs, although a few users reported issues with *Touch ID* when using the phones with dirty fingers; and
- Very few Android users, on the other hand, used the face unlock technique to unlock their phones due to technical difficulties encountered.

The overall conclusions reached by Bhagavatula et al. from their two studies clearly indicate that people more positively perceive the extra security provided by biometrics on their mobile devices compared to traditional methods, such as PINs; furthermore, iPhone's *Touch ID* was determined the most popular biometric.

Android's face unlock mechanism seemed to suffer from some drawbacks that, if fixed, may lead to more large-scale adoption. In general, just as it is important to develop novel biometric authentication techniques for mobile phones, it is equally important to assess user perceptions and attitudes regarding usability in order to make mobile biometric security more prevalent among the masses (Figs. 16.8 and 16.9).

Case Study 2

The use of behavioral biometrics, such as gait and keystroke dynamics, is still not as prevalent in mobile devices as the use of the physical biometrics (e.g., face, fingerprints, and iris). However, existing research indicates that authentication methods can be improved by considering implicit, individual behavioral cues [109, 110]. Verifying identity based on typing behavior—also called "keystroke dynamics"—has been studied thoroughly in the literature with older mobile phones with physical keys [103, 111] as well as with newer devices featuring touchscreens [112, 113].

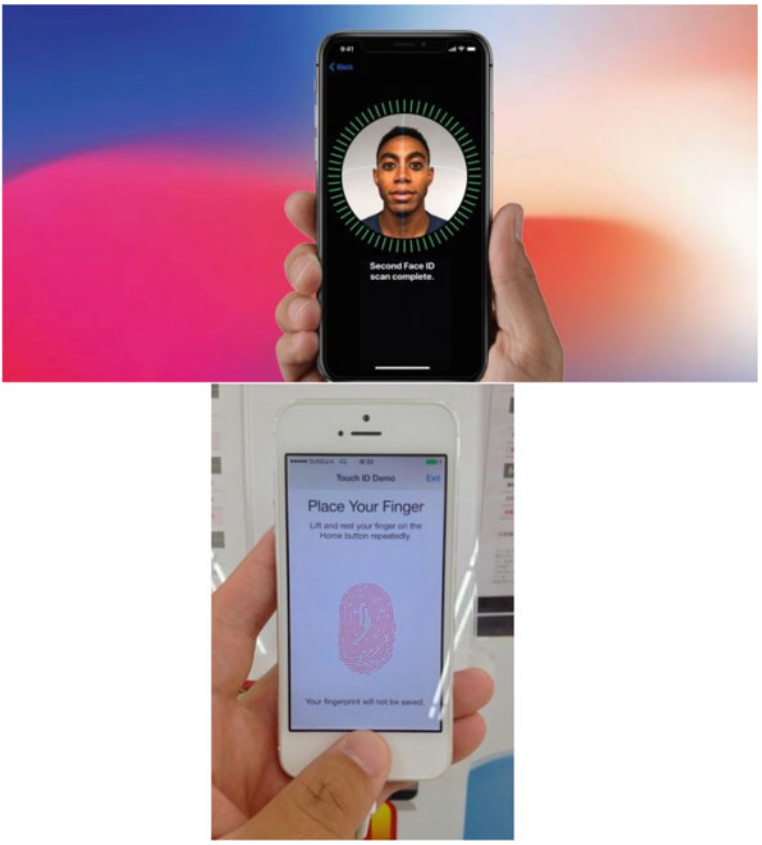

Fig. 16.8 Face ID and Touch ID on Apple iPhones. *Copyright info* Google images—"labeled for reuse"

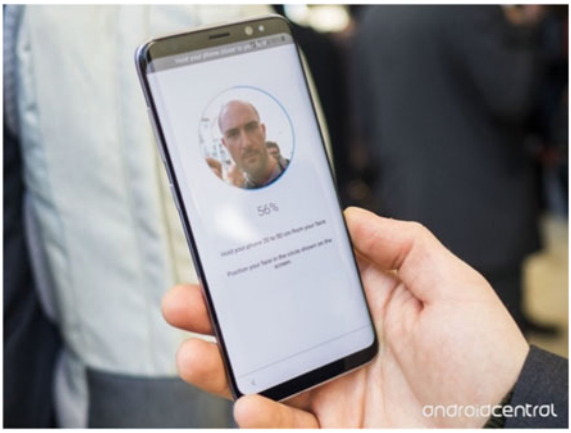

Fig. 16.9 Face unlock on a Samsung Galaxy phone. *Copyright info* Google images—"labeled for reuse"

Buschek et al. [114] presented an in-depth analysis of current keystroke biometrics on current smartphones that provide touch-typing capabilities and included a proposed approach to improve the usability of this method, which we discuss briefly as a case study in this subsection.

Buschek et al. [114] collected data from 28 participants aged an average of 25 years; eight participants were female, twenty were male, and all owned mobile phones with touchscreens and typed with their right hands. Each participant was invited to two sessions that were at least one week apart. Each session comprised three main tasks (three hand postures) and lasted about an hour. For each hand posture, participants typed six different passwords in random order twenty times each. The number of attempts was unlimited, and the user could reenter the password if a wrong attempt was entered during any step.

Some challenges for practical and usable applications of mobile keystroke biometrics are demonstrated by the study's following results:

- The EERs obtained from data collected in a single session were lower than those collected over different sessions, indicating that mobile typing biometrics vary over time;
- Mobile typing biometrics are highly dependent on the specific hand posture; training and testing using multiple postures increased participants' EERs by 86.3% relative to testing with the same hand posture.

These observations imply that an important consideration for improving the usability of mobile keystroke biometrics involves the ability of the application to infer postures dynamically. The latter can be achieved by combining the models for different hand postures using a probabilistic framework that has proven to reduce EERs more significantly than a single model based on one posture. Thus, although using multiple hand postures creates a trade-off between security (lower EERs) and usability, this can be easily addressed (as described above). Since usability is a primary concern for more widespread application of biometrics on the mobile platform,

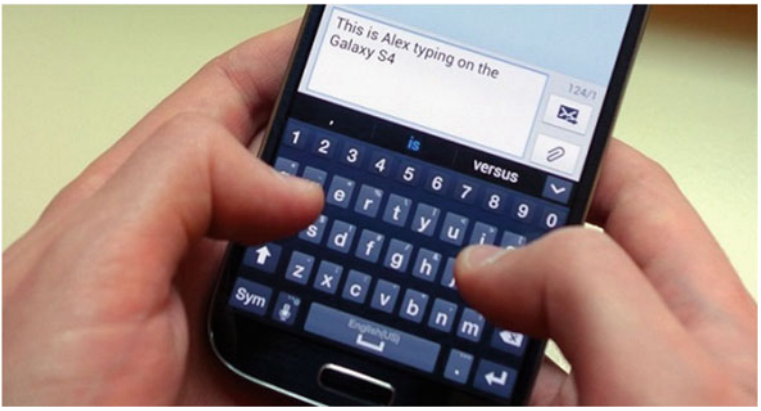

Fig. 16.10 Keystrokes on a Samsung Galaxy phone. *Copyright info* Google images—"labeled for reuse"

this case study offers interesting insights as to how this may be achieved without compromising the level of security attained (Fig. 16.10).

16.7 Conclusion

Users have been eager to embrace selfie biometrics. However, security vulnerabilities and usability issues have emerged. Researchers and mobile device manufacturers have proposed innovative software- and hardware-based techniques meant to overcome these problems, many of which yielded promising results. iPhone X's Face ID system, for example, cannot be deceived with a photograph of the person's face thanks to its imaging technology. However, vulnerabilities continue to pose a threat (e.g., 3-D masks).

As the use of selfie biometrics grows and new modalities find their way onto mobile devices, new security and usability challenges will arise and introduce ripe areas for future innovation and development.

Acknowledgements We would like to thank the students in our research laboratory for their assistance in developing and implementing the approaches discussed in this chapter—specifically, Narciso Sandico, Sadun Muhi, and Eryu Suo. We would additionally like to thank Maria Villa and Bryan Villa for their illustrative work in Fig. 16.5.

References

1. Apple Corporation (2018) IPhone 5s—technical specifications. Retrieved from https://support.apple.com/kb/sp685?locale=en_US. Cited 24 Sept 2018
2. Chaos Computer Club (CCC) (2013) Chaos Computer Club breaks Apple TouchID. Retrieved from https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid. Cited 15 Aug 2018
3. Rogers M (2014) Why I hacked TouchID (again) and still think it's awesome. Lookout Blog. Retrieved from https://blog.lookout.com/iphone-6-touchid-hack. Cited 2 July 2018
4. Cao K, Jain AK (2016) Hacking mobile phones using 2D printed fingerprints. MSU technical report
5. Heisler Y (2017) Security researchers demo how "easy" it is to fool face ID with a 3D mask. BGR. Retrieved from https://bgr.com/2017/11/28/face-id-hack-3d-mask-iphone-x-security/. Cited 2 July 2018
6. Matteson S (2017) IPhone's face ID can be hacked, but here's why nobody needs to panic. TechRepublic. Retrieved from https://www.techrepublic.com/article/iphones-face-id-can-be-hacked-but-heres-why-nobody-needs-to-panic/. Cited 2 July 2018
7. Ratha NK, Connell JH, Bolle RM (2001) Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal 40(3):614–634
8. Apple Corporation (2018) IOS security. Retrieved from https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf. Cited 2 Sept 2018
9. Brandom R (2016) Your phone's biggest vulnerability is your fingerprint. Retrieved from https://www.theverge.com/2016/5/2/11540962/iphone-samsung-fingerprint-duplicate-hack-security. Cited 4 June 2018
10. McKane J (2018) We made fake fingerprints and hacked into a Nokia 5. MyBroadband. Retrieved from https://mybroadband.co.za/news/security/267331-we-made-fake-fingerprints-and-hacked-into-a-nokia-5.html. Cited 3 Aug 2018
11. Smith C (2017) The iPhone X's face ID has one real vulnerability: your kids. BGR. Retrieved from https://bgr.com/2017/11/14/iphone-x-face-id-hacked-children/. Cited 3 July 2018
12. Smith C (2017) Face ID shown unlocking for family members who aren't alike. BGR. Retrieved from https://bgr.com/2017/12/31/iphone-x-face-id-hack-family-members/. Cited 3 July 2018
13. Williams-Grut O (2016) A researcher claims 2 bank apps can be hacked using iPhone's "Live Photos." Business Insider. Retrieved from https://www.businessinsider.com/bank-apps-facial-recognition-hacked-using-iphone-live-photos-2016-8. Cited 3 July 2018
14. International Organization for Standardization (2016) Information technology—biometric presentation attack detection—part 1: framework. Retrieved from https://www.iso.org/standard/53227.html
15. National Institute of Standards and Technology (2013) Standards for biometric technologies. Retrieved from https://www.nist.gov/speech-testimony/standards-biometric-technologies
16. Galbally J, Marcel S, Fierrez J (2014) Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition. IEEE Trans Image Process 23(2):710–724
17. Rattani A, Derakhshani R (2018) A survey of mobile face biometrics. Comput Electr Eng 72:39–52
18. Amadeo R (2017) Galaxy S8 face recognition already defeated with a simple picture. Ars Technica. Retrieved from https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/. Cited 3 July 2018
19. Moren D (2015) Face recognition security, even with a "blink test," is easy to trick. Popular Science. Retrieved from https://www.popsci.com/its-not-hard-trick-facial-recognition-security?utm_medium=twitter&utm_source=twitterfeed. Cited 3 July 2018
20. de Freitas Pereira T, Anjos A, De Martino JM, Marcel S (2013) Can face anti spoofing countermeasures work in a real world scenario? Paper presented at the IEEE international conference on biometrics (ICB). Madrid, Spain
21. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recogn Lett 15(11):1119–1125

22. Erdogmus N, Marcel S (2013) Spoofing 2D face recognition systems with 3D masks. Paper presented at the 2013 international conference of the biometrics special interest group (BIOSIG), Darmstadt, Germany
23. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
24. Li J, Wang Y, Tan T, Jain AK (2004) Live face detection based on the analysis of fourier spectra. Biomet Technol Hum Ident 5404:296–304
25. Cardinal D (2017) How Apple's iPhone X TrueDepth camera works. ExtremeTech. Retrieved from https://www.extremetech.com/mobile/255771-apple-iphone-x-truedepth-camera-works. Cited 14 Sept 2017
26. Apple Corporation (2017) Face ID security. Retrieved from https://www.apple.com/business/site/docs/FaceID_Security_Guide.pdf
27. InAuth (2017) Fingerprints: the most popular biometric. Retrieved from https://www.inauth.com/blog/fingerprints-popular-biometric/. Cited 2 July 2018
28. Tess (2017) Realistic 3D printed finger could make smartphone fingerprint scanners harder to hack. 3drs.org. Retrieved from https://www.3ders.org/articles/20170925-realistic-3d-printed-finger-could-make-smartphone-fingerprint-scanners-harder-to-hack.html
29. Marasco E, Ross A (2015) A survey on antispoofing schemes for fingerprint recognition systems. ACM Comput Surv (CSUR) 47(2):28
30. Bowden-Peters E, Phan RCW, Whitley JN, Parish DJ (2012) Fooling a liveness-detecting capacitive fingerprint scanner. Cryptography and security: from theory to applications. Springer, Berlin, pp 484–490
31. Phone Arena (2018) Samsung Galaxy S9 vs Samsung Galaxy S5—Phone specs comparison. Retrieved from https://www.phonearena.com/phones/compare/Samsung-Galaxy-S9,Samsung-Galaxy-S5/phones/10717,8202. Cited 3 July 2018
32. Fox-Brewster T (2018) Yes, cops are now opening iPhones with dead people's fingerprints. Forbes. Retrieved from https://www.forbes.com/sites/thomasbrewster/2018/03/22/yes-cops-are-now-opening-iphones-with-dead-peoples-fingerprints/#3e50d3c7393e. Cited 3 July 2018
33. Hardy E (2018) Cops will use Touch ID on your corpse to unlock your iPhone. Cult of Mac. Retrieved from https://www.cultofmac.com/536691/police-unlock-iphones-with-dead-fingers-touch-id/. Cited 4 July 2018
34. Wehner M (2016) Why a disembodied finger can't be used to unlock the touch ID sensor on the iPhone 5s. Engadget. Retrieved from https://www.engadget.com/2013/09/16/why-a-disembodied-finger-cant-be-used-to-unlock-the-touch-id-se/. Cited 4 July 2018
35. Etherington D (2013). Watch a cat unlock the iPhone 5s using touch ID and the fingerprint sensor. Retrieved from https://techcrunch.com/2013/09/19/watch-a-cat-unlock-the-iphone-5s-using-touch-id-and-the-fingerprint-sensor/. Cited 4 July 2018
36. Leopold T (2013) New iPhone 5S fingerprint sensor works for dogs. CNN. Retrieved from https://www.cnn.com/2013/09/20/tech/mobile/iphone-dog-paw-print-ireport/index.html. Cited 3 July 2018
37. Kooser A (2016) See a hedgehog unlock an iPhone with its tiny paw. CNET. Retrieved from https://www.cnet.com/news/hedgehog-unlock-iphone-sashimi/. Cited 4 July 2018
38. Qualcomm (2017) Qualcomm fingerprint sensors. Retrieved from https://www.qualcomm.com/solutions/mobile-computing/features/security/fingerprint-sensors. Cited 4 July 2018
39. Qualcomm (2018) Qualcomm announces advanced fingerprint scanning and authentication technology. Retrieved from https://www.qualcomm.com/news/releases/2017/06/28/qualcomm-announces-advanced-fingerprint-scanning-and-authentication. Cited 4 July 2018
40. Avila CS, Casanova JG, Ballesteros F, Garcia LRT, Gomez MFA, Sierra DS (2014) State of the art of mobile biometrics, liveness and non-coercion detection. Personalized Centralized Authentication System
41. Abhyankar A, Schuckers S (2006) Fingerprint liveness detection using local ridge frequencies and multiresolution texture analysis techniques. Paper presented at the IEEE international conference on image processing (ICIP). Atlanta, GA

42. Jiao J, Deng Z (2017) Deep combining of local phase quantization and histogram of oriented gradients for indoor positioning based on smartphone camera. Int J Distrib Sens Netw 13(1):1550147716686978
43. Coli P, Marcialis G, Roli F (2007) Power spectrum-based fingerprint vitality detection. Paper presented at the IEEE international work on automatic identification advanced technologies (AutoID). Alghero, Italy
44. Six J, Cornelis O, Leman M (2014) TarsosDSP, a real-time audio processing framework in Java. Paper presented at the audio engineering society 53rd international conference: semantic audio. London, England
45. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
46. Muhammad A (2015) OpenCV Android programming by example. Packt Publishing Ltd, Birmingham
47. Nikam SB, Agarwal S (2008) Texture and wavelet-based spoof fingerprint detection for fingerprint biometric systems. Paper presented at the 2018 first international conference on emerging trends in engineering and technology, Nagpur, Maharashtra, India
48. Jia X, Yang X, Cao K, Zang Y, Zhang N, Dai R, Tian J (2014) Multi-scale local binary pattern with filters for spoof fingerprint detection. Inf Sci 268:91–102
49. Yambay D, Ghiani L, Denti P, Marcialis GL, Roli F, Schuckers S (2012) LivDet 2011—Fingerprint liveness detection competition 2011. Paper presented at the 2012 5th IAPR international conference on biometrics (ICB), New Delhi, India
50. Kumpituck S, Li D, Kunieda H, Isshiki T (2017) Fingerprint spoof detection using wavelet based local binary pattern. Paper presented at the 8th international conference on graphic and image processing (ICGIP 2016). Bellingham, WA
51. Kumar L, Sharma K (2013) Web based novel technique for watermarking colour images on Android mobile phones. Int J Adv Res Comput Sci Softw Eng 3(7)
52. Gragnaniello D, Poggi G, Sansone C, Verdoliva L (2013) Fingerprint liveness detection based on weber local image descriptor. Paper presented at the 2013 IEEE workshop on biometric measurements and systems for security and medical applications (BIOMS). Naples, Italy
53. Kannala J, Rahtu E (2012) Bsif: binarized statistical image features. Paper presented at the 2012 21st international conference on pattern recognition (ICPR). Tsukuba, Japan
54. Superpowered (n.d.) IOS and Android FFT & iOS and Android Polar FFT. Retrieved from https://superpowered.com/fft-and-polar-fft. Cited 4 July 2018
55. Manivanan N, Memon S, Balachandran W (2010) Automatic detection of active sweat pores of fingerprint using highpass and correlation filtering. Electron Lett 46(18):1268–1269
56. Manivanan N, Memon S, Balachandran W (2010) Security breaks a sweat. Electron Lett 46(18):1241–1242
57. Espinoza M, Champod C (2011) Using the number of pores on fingerprint images to detect spoofing attacks. Paper presented at the 2011 international conference on hand-based biometrics (ICHB), Hong Kong, China
58. Marcialis GL, Roli F, Tidu A (2010) Analysis of fingerprint pores for vitality detection. Paper presented at the 2010 20th international conference on pattern recognition (ICPR). Istanbul, Turkey
59. Memon SA (2012) Novel active sweat pores based liveness detection techniques for fingerprint biometrics. Doctoral dissertation. Brunel University School of Engineering and Design Ph.D. theses
60. Memon S, Manivannan N, Balachandran W (2011) Active pore detection for liveness in fingerprint identification system. Paper presented at the 2011 19th telecommunications forum (TELFOR), Belgrade, Serbia
61. Rattani A, Scheirer WJ, Ross A (2015) Open set fingerprint spoof detection across novel fabrication materials. IEEE Trans Inf Forensics Secur 10(11):2447–2460
62. Rattani A, Ross A (2014) Automatic adaptation of fingerprint liveness detector to new spoof materials. Paper presented at the IEEE international joint conference on biometrics. Clearwater, FL

63. Rattani A, Ross A (2014a) Minimizing the impact of spoof fabrication material on fingerprint liveness detector. Paper presented at the 2014 IEEE international conference on image processing (ICIP). Paris, France
64. Bhide B (2013) Low-pass-filter-to-Android-sensors. Retrieved from https://github.com/Bhide/Low-Pass-Filter-To-Android-Sensors. Cited 4 July 2018
65. W3C Working Group (2017) Motion sensors explainer. Retrieved from https://www.w3.org/TR/motion-sensors/#low-pass-filters. Cited 4 July 2018
66. Alzantot M, Wang Y, Ren Z, Srivastava MB (2017) RSTensorFlow: GPU enabled TensorFlow for deep learning on commodity android devices. Paper presented at the 1st international workshop on deep learning for mobile systems and applications. Niagara Falls, NY
67. Google Corporation (n.d.) Change "Ok Google" settings. Retrieved from https://support.google.com/assistant/answer/7394306?hl=en. Cited 4 July 2018
68. Young PJ, Jin JH, Woo S, Lee DH (2016) BadVoice: soundless voice-control replay attack on modern smartphones. Paper presented at the 2016 eighth international conference on ubiquitous and future networks (ICUFN). Vienna, Austria
69. Richardson M, Wallace S (2012) Getting started with raspberry PI. O'Reilly Media Inc., Sebastopol
70. Wu Z, Evans N, Kinnunen T, Yamagishi J, Alegre F, Li H (2015) Spoofing and countermeasures for speaker verification: a survey. Speech Commun 66:130–153
71. Pew Research Center (2018) Demographics of mobile device ownership and adoption in the United States. Retrieved from http://www.pewinternet.org/fact-sheet/mobile/. Cited 3 July 2018
72. Shang W, Stevenson M (2010) Score normalization in playback attack detection. Paper presented at the IEEE international conference on acoustics, speech, and signal processing (ICASSP). Dallas, TX
73. Villalba J, Lleida E (2011) Detecting replay attacks from far-field recordings on speaker verification systems. European workshop on biometrics and identity management. Springer, Berlin, pp 274–285
74. Wang ZF, Wei G, He QH (2011) Channel pattern noise based playback attack detection algorithm for speaker recognition. Paper presented at the 2011 international conference on machine learning and cybernetics (ICMLC). Guilin, China
75. Rossi M, Feese S, Amft O, Braune N, Martis S, Tröster G (2013) AmbientSense: a real-time ambient sound recognition system for smartphones. Paper presented at the 2013 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops). San Diego, CA
76. Chen LW, Guo W, Dai LR (2010) Speaker verification against synthetic speech. Paper presented at the 7th international symposium on Chinese spoken language processing (ISCSLP). Tainan, Taiwan
77. Wu Z, Chng ES, Li H (2012) Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. Interspeech
78. Wu Z, Kinnunen T, Chng ES, Li H, Ambikairajah E (2012) A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case. Paper presented at the 2012 Asia-Pacific signal information processing association annual summit and conference (APSIPA ASC). Hollywood, CA
79. Alegre F, Amehraye A, Evans N (2013) A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. Paper presented at the international conference on biometrics: theory, applications and systems (BTAS). Alrington, VA
80. Alegre F, Amehraye A, & Evans N (2013) Spoofing countermeasures to protect automatic speaker verification from voice conversion. Paper presented at the IEEE international conference on acoustics, speech, and signal processing (ICASSP). Vancouver, BC
81. Matrouf D, Bonastre JF, Fredouille C (2006) Effect of speech transformation on impostor acceptance. Paper presented at the 2006 IEEE international conference on acoustics, speech, and signal processing (ICASSP). Toulouse, France

82. Gofman M, Sandico N, Mitra S, Suo E, Muhi S, Vu T (2018) Multimodal biometrics via discriminant correlation analysis on mobile devices. Paper presented at the 2018 international conference on security and management. Las Vegas, NV
83. Chen S, Ren K, Piao S, Wang C, Wang Q, Weng J, Su L, Mohaisen A (2017) You can hear but you cannot steal: defending against voice impersonation attacks on smartphones. Paper presented at the 2017 IEEE 37th international conference on distributed computing systems (ICDCS). Atlanta, GA
84. Khoury E, El Shafey L, Marcel S (2014) Spear: an open source toolbox for speaker recognition based on Bob. Paper presented at the 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). Florence, Italy
85. Kominek J, Black AW (2004) The CMU arctic speech databases. Paper presented at the 5th ISCA workshop on speech synthesis. Pittsburgh, PA
86. Feng H, Fawaz K, Shin KG (2017) Continuous authentication for voice assistants. Paper presented at the 23rd annual international conference on mobile computing and networking. Snowbird, UT
87. Zhang L, Tan S, Yang J (2017) Hearing your voice is not enough: an articulatory gesture based liveness detection for voice authentication. Paper presented at the 2017 ACM SIGSAC conference on computer and communications security. Dallas, TX
88. Zhang L, Tan S, Yang J, Chen Y (2016) Voicelive: a phoneme localization based liveness detection for voice authentication on smartphones. Paper presented at the 2016 ACM SIGSAC conference on computer and communications security. Vienna, Austria
89. Rodrigues RN, Ling LL, Govindaraju V (2009) Robustness of multimodal biometric fusion methods against spoof attacks. J Vis Lang Comput 20(3):169–179
90. Gofman MI, Mitra S, Smith N (2016) Hidden Markov models for feature-level fusion of biometrics on mobile devices. Paper presented at the 2016 IEEE/ACS 13th international conference of computer systems and applications (AICCSA). Agadir, Morocco
91. Gofman MI, Mitra S, Cheng THK, Smith NT (2016) Multimodal biometrics for enhanced mobile device security. Commun ACM 59(4):58–65
92. Katona M et al (2005) FPGA design and implementation of a wavelet-domain video denoising system. Lect Notes Comput Sci 3708:650–657
93. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Handbook of fingerprint recognition, 2nd edn. Springer, New York City
94. Teoh ABJ, Goh A, Ngo DCL (2006) Random multispace quantization as an analytic mechanism for BioHashing of biometric and random identity inputs. IEEE Trans Pattern Anal Mach Intell 28(12):1892–1901
95. Arjona R, Baturone I (2015) A fingerprint biometric cryptosystem in FPGA. Paper presented at the 2015 IEEE international conference on industrial technology (ICIT). Seville, Spain
96. Imamverdiyev Y, Teoh ABJ, Kim J (2013) Biometric cryptosystem based on discretized fingerprint texture descriptors. Expert Syst Appl 40(5):1888–1901
97. Pepicq B (2017) Why do some people refuse to use Google Assistant? AndroidPIT. Retrieved from https://www.androidpit.com/why-not-use-google-assistant. Cited 3 June 2018
98. Mohsin MA (2017) An FPGA-based hardware accelerator for *k*-nearest neighbor classification for machine learning. Master thesis. University of Colorado Springs
99. Paul R (2011) Unwrapping a new ice cream sandwich: Android 4.0 reviewed. Ars Technica. Retrieved from https://arstechnica.com/gadgets/2011/12/unwrapping-a-new-ice-cream-sandwich-android-40-reviewed-1/. Cited 20 June 2018
100. Apple (2018) Using touch ID on the iPhone. Retrieved from http://support.apple.com/kb/ht5883. Cited 20 June 2018
101. Chamary JV (2017) No, Apple's face ID is not a "secure Password". Forbes. Retrieved from https://www.forbes.com/sites/jvchamary/2017/09/18/security-apple-face-id-iphone-x/#99580fc4c835. Cited 28 June 2018
102. Bhagavatula C, Ur B, Iacovino K, Kywe SM, Cranor LF, Savvides M (2015) Biometric authentication on iPhone and Android: usability, perceptions, and influences on adoption. Paper presented at the usable security (USEC). Workshop. San Diego, CA

103. Clarke NL, Furnell SM (2006) Authenticating mobile phone users using keystroke analysis. Int J Inf Secur 6(1):1–14
104. Trewin S, Swart C, Koved L, Martino J, Singh K, Ben-David S (2012) Biometric authentication on a mobile device: a study of user effort, error and task disruption. Paper presented at the 28th annual computer security applications conference (ACSAC). Orlando, FL
105. Braz C, Robert J-M (2006) Security and usability: the case of the user authentication methods. Paper presented at the 18th International conference of the association francophone d'Interaction Homme-Machine. Montreal, Quebec
106. Egelman S, Jain S, Portnoff RS, Liao K, Consolvo S, Wagner D (2014) Are you ready to lock? Paper presented at the ACM SIGSAC conference on computer & communications security. Scottsdale, AZ
107. Harbach M, von Zezschwitz E, Fichtner A, De Luca A, Smith M (2014) It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception. Paper presented at the symposium on usable privacy and security. Menlo Park, CA
108. Sathiah S (2017) Face ID on the iPhone X is a backwards step in usability. Notebookcheck. Retrieved from https://www.notebookcheck.net/Face-ID-on-the-iPhone-X-is-a-backwards-step-in-usability.264306.0.html. Cited 28 June 2018
109. Burgbacher U, Hinrichs K (2014) An implicit author verification system for text messages based on gesture typing biometrics. Paper presented at the ACM CHI conference on human factors in computing systems. Toronto, Canada
110. Crawford H (2010) Keystroke dynamics: characteristics and opportunities. In: 8th international conference on privacy, security and trust. Ottawa, CA
111. Campisi P, Maiorana E, Lo Bosco M, Neri A (2009) User authentication using keystroke dynamics for cellular phones. IET Sig Process 3(5):333–341
112. Nauman M, Ali T, Rauf A (2013) Using trusted computing for privacy preserving keystroke-based authentication in smartphones. Telecommun Syst 52:2149–2161
113. Saevanee H, Bhattarakosol P (2009) Authenticating user using keystroke dynamics and finger pressure. Paper presented at the 6th IEEE consumer communications and networking conference. Las Vegas, NV
114. Buschek D, De Luca A, Alt F (2015) Improving accuracy, applicability, and usability of keystroke biometrics on mobile touchscreen devices. Paper presented at the ACM CHI 2015 conference, crossings. Seoul, Korea